ArcticLink Solution Platform User Manual

Revision B

Contact Information

QuickLogic Corporation
1277 Orleans Drive
Sunnyvale, CA 94089
Phone: (408) 990-4000 (US)
(905) 940-4149 (Canada)
+(44) 1932 57 9011 (Europe)
+(86) 21 6867 0273 (Asia – except Japan)
+(81) 45 470 5525 (Japan)
E-mail: info@quicklogic.com
Sales: www.quicklogic.com/sales
Support: www.quicklogic.com/support
Internet: www.quicklogic.com

Revision History

Revision	Date	Originator and Comments
A	March 2007	Initial version
B	July 2007	Updated Data Flow and Programmable Fabric chapters, added Design Considerations chapter – Jason Lew and Elaine Chan

Copyright and Trademark Information

Copyright © 2007 QuickLogic® Corporation.
All Rights Reserved.

The information contained in this document is protected by copyright. All rights are reserved by QuickLogic Corporation. QuickLogic Corporation reserves the right to modify this document without any obligation to notify any person or entity of such revision. Copying, duplicating, selling, or otherwise distributing any part of this product without the prior written consent of an authorized representative of QuickLogic is prohibited.

QuickLogic and the QuickLogic logo, ArcticLink and QuickWorks are trademarks or registered trademarks of QuickLogic Corporation. I^2C is a trademark of Philips Corporation.

All other trademarks or registered trademarks are the properties of their respective owners.

Contents

Chapter 1: Getting Started 1

1.1 About This Manual 1
1.1.1 Purpose 1
1.1.2 Who Should Read This Manual 1
1.2 How to Use This Manual 1
1.3 Getting Help from QuickLogic 2
1.4 Getting Answers to USB Related Questions 2
1.5 Getting Answers to SD Related Questions 2
1.6 Disclaimer 2

Chapter 2: Introduction 3

2.1 Product Overview 3
2.2 ArcticLink Solution Platform Features 5
2.2.1 Hi-Speed USB 2.0 OTG 5
2.2.2 SD/SDIO/MMC/CE-ATA 6
2.2.3 Library of Interfaces in Programmable Fabric 6
2.2.4 Low Power Programmable Fabric 6
2.2.5 Lead-Free Packaging 6
2.2.6 Supported Processors 6

Chapter 3: Data Flow 7

3.1 Functional Overview 7
3.1.1 Hi-Speed USB 2.0 OTG Architecture 10
3.1.2 SD/SDIO/MMC/CE-ATA Architecture 11
3.1.3 Progammable Fabric 12
3.1.4 Dual-Port Scratchpad 8 KB SRAM 13
3.1.5 Fast Peripheral Bus Interface 13
3.2 USB OTG Controller and Fabric Data Flow 13
3.2.1 Host Mode Data Flow 13
3.2.2 Device Mode Data Flow 16
3.3 SD/SDIO/MMC/CE-ATA Controller and Fabric Data Flow 18
3.3.1 Host Mode Data Flow 18
3.4 Datapath and Usage Models 20
3.4.1 SDIO/SD/CE-ATA Datapath and Usage Model 21
3.4.2 USB Datapath Usage Models 22

Chapter 4: Hi-Speed USB 2.0 OTG Controller 25

© 2007 QuickLogic Corporation

4.1 Functional Overview 25
4.2 Features 27
4.3 Operation 28
4.3.1 USB 2.0 OTG Controller Initialization 28
4.3.2 Core Initialization 28
4.3.3 Operating Modes 29
4.3.4 Host Mode Operation in USB Transactions 29
4.3.5 Device Mode Operation In USB Transactions 31
4.3.6 Programming Models 32

Chapter 5: SD/SDIO/MMC/CE-ATA Controller 35

5.1 Functional Overview 35
5.1.1 SDIO System Interface 35
5.2 Features 35
5.3 SD/SDIO/MMC/CE-ATA Host Controller 36
5.3.1 Host Registers 37
5.3.2 Data FIFO 38
5.3.3 Dynamic Buffer Management 38
5.3.4 CMD Control 38
5.3.5 DAT Control 39
5.3.6 Interrupt Control Logic 41
5.3.7 Clock Management 41
5.3.8 SDIO Power Management and Card Detection 42
5.3.9 SD/SDIO/MMC/CE-ATA Host Controller Interface 42
5.3.10 CE-ATA Initialization Sequence 43

Chapter 6: Programmable Fabric 45

6.1 General Programmable Fabric Features 45
6.2 ASSP/Programmable Fabric Interface Ports 46
6.2.1 Multi-Function Pins 46
6.3 Designing with ArcticLink in QuickWorks 47
6.3.1 File Types 47
6.3.2 Design using Schematics 48
6.3.3 Design using Verilog/VHDL 49
6.3.4 Synthesizing ArcticLink Programmable Fabric Design 50
6.4 Simulation Modeling 51
6.4.1 ASSP/Fabric Interfaces 52
6.4.2 Slave Interface 53
6.4.3 Master Interface 55
6.4.4 Pre-layout Simulation 56
6.4.5 Post-layout Simulation 56

© 2007 QuickLogic Corporation

Chapter 7: Design Considerations 57

7.1 ArcticLink Design Interfaces 57
7.1.1 ASSP/Programmable Fabric Interface Ports 57
7.1.2 Clocking Schemes and CCM Usage 57
7.1.3 Peripheral vs. Fast Peripheral Interface 58
7.1.4 Top-level Multi-function Pins 58
7.2 Application Models 59
7.2.1 Marvell Monahan DFI Interface 59
7.2.2 Memory DMA and Bus Mastering 59
7.2.3 USB Enumeration 60
7.2.4 USB Hub 61
7.2.5 Carkit Support in Programmable Fabric 62
7.2.6 ULPI PHY Impersonator in Programmable Fabric 63

Chapter 8: Control and Status Registers 65

8.1 ArcticLink System 66
8.1.1 Memory Map 66
8.1.2 Common Registers 67
8.2 SD/SDIO/MMC and CE–ATA Register Descriptions 73
8.2.1 SD/SDIO/MMC/CE-ATA Host Controller Register Set 73
8.2.2 SD/SDIO and CE–ATA Registers 73
8.2.3 Host Control Register (Offset 28h) 74
8.2.4 Normal Interrupt Status Register (Offset 30h) 74
8.2.5 Normal Interrupt Status Enable Register (Offset 34h) 75
8.2.6 Normal Interrupt Signal Enable Register (Offset 38h) 75
8.2.7 Capability Register (Offset 40h) 75
8.2.8 Maximum Current Capabilities Register (Offset 48h) 76
8.2.9 CE-ATA Control Register (Offset 100h) 77
8.3 USB OTG Control and Status Registers 77
8.3.1 USB OTG Control and Status Overview 77
8.3.2 CSR Memory Map 78
8.3.3 Register Descriptions 86

© 2007 QuickLogic Corporation

© 2007 QuickLogic Corporation

Chapter 1
Getting Started

This chapter provides tips on how to use this manual, how to get more information on related products and technologies, and how to contact QuickLogic with inquiries about the device. It contains the following sections:

- **"About This Manual" on page 1**
- **"How to Use This Manual" on page 1**
- **"Getting Help from QuickLogic" on page 2**
- **"Getting Answers to USB Related Questions" on page 2**
- **"Getting Answers to SD Related Questions" on page 2**
- **"Disclaimer" on page 2**

1.1 About This Manual

1.1.1 Purpose

The QuickLogic ArcticLink Solution Platform User Manual provides detailed information on QuickLogic's ArcticLink solution platform designed for portable electronics.

1.1.2 Who Should Read This Manual

This manual is intended for software and hardware engineers, product managers, and other technical staff involved in the design and implementation of Portable Electronics. It assumes advanced knowledge of electrical engineering terminology or access to that knowledge.

1.2 How to Use This Manual

The latest versions of the QuickLogic ArcticLink Solution Platform User Manual and other complementary documents are available in PDF format on the QuickLogic website (www.quicklogic.com). A complete documentation set for the ArcticLink solution platform includes the following:

- QuickLogic's *ArcticLink Solution Platform User Manual* (this document).
- QuickLogic's *ArcticLink Solution Platform Data Sheet* (contains timing information and electrical specifications).
- QuickLogic's *ArcticLink Solution Platform Product Brief* (contains product overview and examples for applications).
- *Universal Serial Bus Revision 2.0 Specification* (contains guidelines for designing USB systems and peripherals).

© 2007 QuickLogic Corporation

- *Simplified Version of the SD Host Controller Specification* (contains guidelines for designing SD Host Controllers).

Before finalizing a system design based on the ArcticLink devices, please contact QuickLogic to verify that you have the most recent specifications. The latest application notes and design aids can be found on the QuickLogic website.

QuickLogic is constantly trying to improve the quality of its product documentation. If you have any questions or comments, please contact QuickLogic Technical Support at http://www.quicklogic.com/support.

1.3 Getting Help from QuickLogic

You can contact QuickLogic in one of the following ways:

- Online technical support is available at http://www.quicklogic.com/support.
- The most up-to-date technical support information is also posted on the QuickLogic website (www.quicklogic.com).

1.4 Getting Answers to USB Related Questions

The *Universal Serial Bus Revision 2.0 Specification* published by the USB Implementers Forum, Inc. provides more familiarity with the industry-standard USB. However, this manual assumes you already possess an understanding of the industry-standard USB used to design and build systems or peripherals compliant with this standard.

If you are looking for a copy of the Specification in which this information is available, please visit the USB Implementers Forum at their website www.usb.org/developers/docs/.

1.5 Getting Answers to SD Related Questions

This manual assumes a basic understanding of the standard register set to control SD memory cards and SDIO cards. If you are looking for a copy of the Specification in which this information is available, please visit the SD Card Association at their website www.sdcard.org/HostController/index.html.

If you are not very familiar with the SD standard register set, a good place to start is by downloading and reading the *SD Host Controller Standard Simplified Specification* published by the SD Card Association.

1.6 Disclaimer

QuickLogic makes no warranties for the use of its products. QuickLogic does not assume any liability for errors which may appear in this document. However, we attempt to notify customers of such errors. QuickLogic retains the right to make changes to either the documentation, specification, or component without notice.

Please verify with QuickLogic that you have the latest specifications before finalizing your design.

© 2007 QuickLogic Corporation

Chapter 2
Introduction

This chapter presents an overview of the ArcticLink architecture and a description of its main features. It contains the following sections:

- **"Product Overview" on page 3**
- **"ArcticLink Solution Platform Features" on page 5**

2.1 Product Overview

The ArcticLink family of programmable connectivity platforms combines fixed interconnect building blocks such as Hi-Speed (HS) USB 2.0 On-The-Go (OTG) and SD/SDIO/MMC/CE-ATA Host Controllers with an embedded ultra-low power programmable fabric block in its single-chip solution.

Hi-Speed USB 2.0 OTG and SD/SDIO/MMC/CE-ATA controllers including the 8 KB SRAM are fixed function logic and constitute the Application Specific Standard Product (ASSP) portion of the device.

The ArcticLink family combines ultra-low power with small form factor packaging and integrated multiple interfaces to solve the emerging needs of mobile consumer electronics connectivity. The flexibility of the ArcticLink platform allows easy system architectural designs and trade-offs.

The ArcticLink solution platform provides the following advantages:

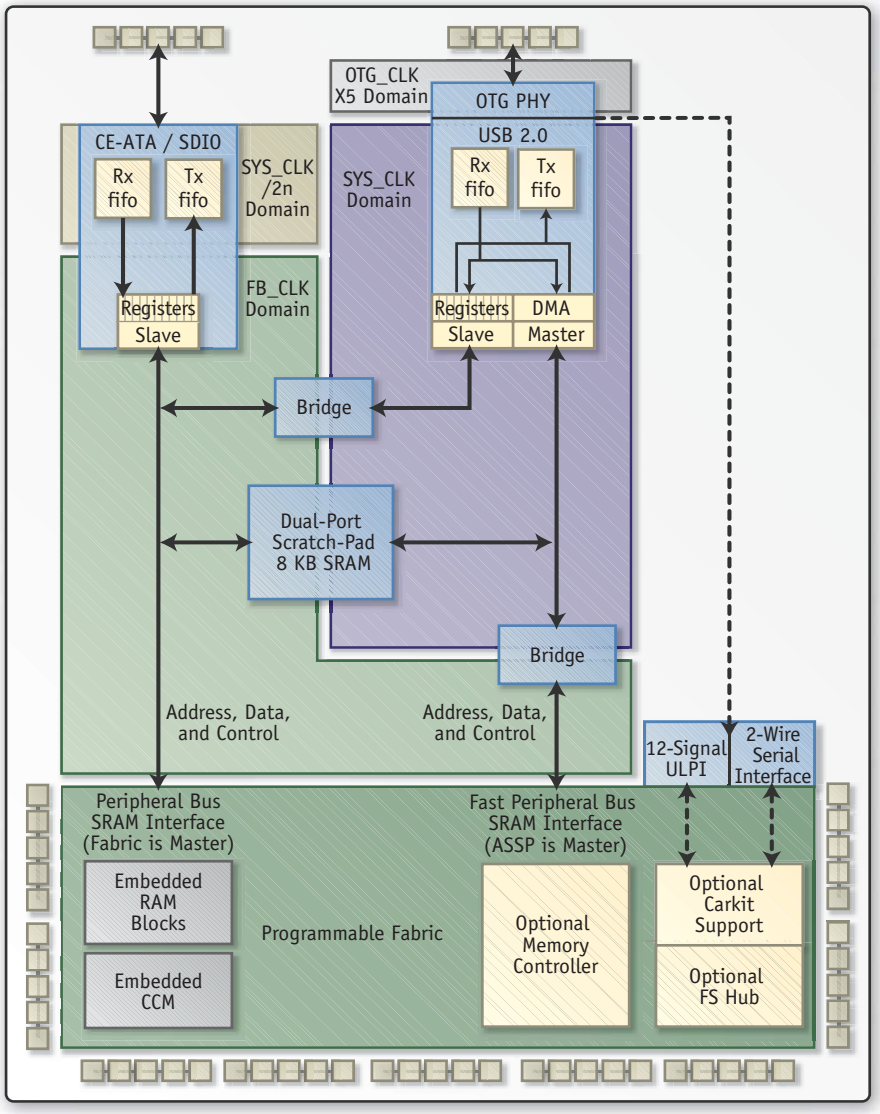
- Integrated, single-chip solution for multiple Host Controllers and interfaces to reduce both system BOM cost and board space.
- Flexible platform with configurable Hi-Speed USB 2.0 OTG, high-speed SDIO, multi-format storage solutions, programmable fabric for additional connectivities and custom functions.
- Fast time-to-market with QuickLogic's complete low power connectivity solutions.

The ArcticLink solution platform interfaces with application processors that have multiplexed or de-multiplexed local bus.

Figure 2-1 presents the ArcticLink platform top level block diagram. Along with the hard-wired Hi-Speed USB 2.0 OTG, SD/SDIO/MMC/CE-ATA blocks, and the programmable Fabric can be used to implement additional interfaces such as PCI, IDE, NAND controller, Bluetooth 2.0 UART, and SPI. The innovative internal split bus architecture allows sustained and concurrent data transfers between different interfaces.

© 2007 QuickLogic Corporation

Figure 2-1: ArcticLink Solution Platform Top Level Block Diagram



The ArcticLink development environment enables concurrent software and hardware development. This environment includes an ArcticLink Solution Platform with software drivers and an ArcticLink system model. With the ArcticLink Solution Platform, software engineers can debug their code while hardware engineers are evaluating the trade-off between what should be implemented in software versus hardware. In addition, the software engineers can also start emulating the system using the ArcticLink Solution Platform.

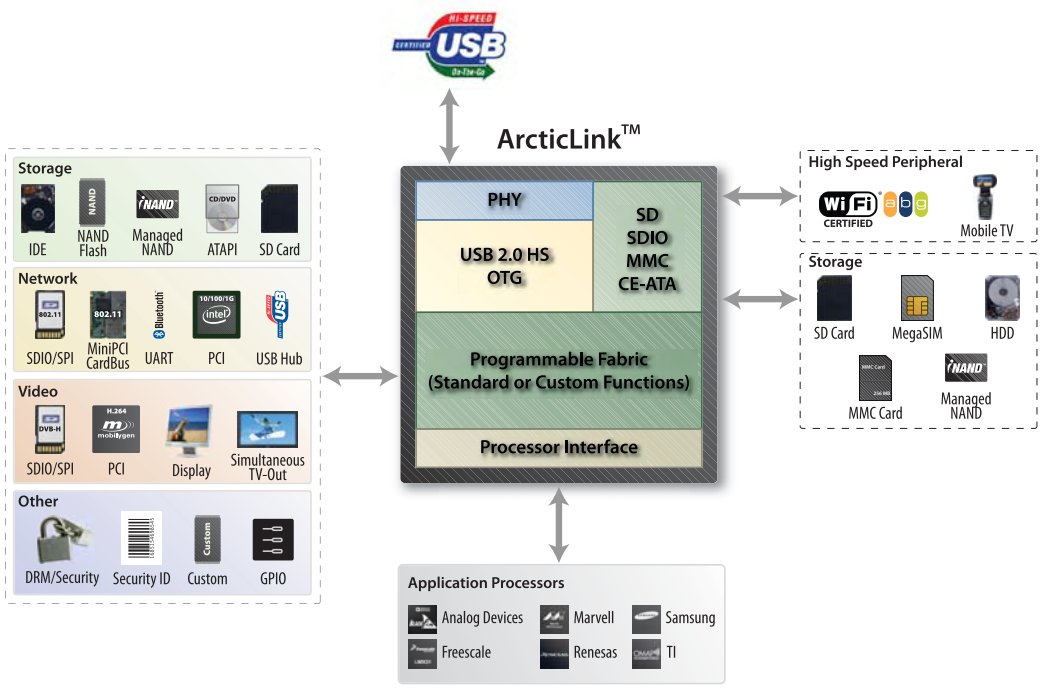
© 2007 QuickLogic Corporation

2.2 ArcticLink Solution Platform Features

This section summarizes key features of the ArcticLink family.

The ArcticLink Solution Platform consists of three blocks — Hi-Speed USB 2.0 OTG, SDIO/SD/MMC/CE-ATA, and programmable fabric. **Figure 2-2** provides a visual overview of the ArcticLink's features and connectivity solutions. The programmable fabric can be altered to suit specific system interconnect requirements in a range of applications around a variety of processor and processorless systems.

Figure 2-2: ArcticLink Connectivity Solutions



2.2.1 Hi-Speed USB 2.0 OTG

- Single port Hi-Speed USB 2.0 OTG controller with integrated PHY
- Supports HS 480 Mb/s, FS 12 Mb/s, and LS 1.5 Mb/s operation
- Dedicated DMA controller
- 12-signal ULPI interface available through programmable fabric
- Supports full-speed CEA-936-A, I2C compatible serial bus carkit interface.
  - For more information about CEA-936-A, see the CE Association website www.ce.org/Standards/StandardDetails.aspx?Id=1416&number=CEA-936-A.
- High-speed connectivity to PC, Wi-Fi, HSDPA and WiMAX

© 2007 QuickLogic Corporation

2.2.2 SD/SDIO/MMC/CE-ATA

- Supports SD 2.0, MMC and CE-ATA 1.10
- 1-bit, 4-bit and 8-bit operation at speeds up to 52 MHz
- 52 MB/s maximum transfer rate
- Support for DMA engine in programmable fabric

2.2.3 Library of Interfaces in Programmable Fabric

- PCI, Mini-PCI, and CardBus
- IDE for HDD and DVD-ROM
- NAND Flash and Solid State Drives
- SDIO and SPI for Wi-Fi and Mobile TV (DVB, T-DMB, ISDB-T)
- High-speed UART for Bluetooth 2.0 and GPS
- ULPI-based PHY-less system interconnect

2.2.4 Low Power Programmable Fabric

- User programmable fabric for custom functions
- Flexible host CPU interface
- An innovative split bus architecture for sustained and concurrent data transfer

2.2.5 Lead-Free Packaging

- 121-ball 8 mm x 8 mm, 0.65 mm pitch CTBGA
- 196-ball 12 mm x 12 mm, 0.8 mm pitch TFBGA
- Known good die availability for multi-die and SIP formats

2.2.6 Supported Processors

Processors with a multiplexed or de-multiplexed local bus:

- Marvell PXA2xx and PXA3xx
- Freescale i.MX Application Processors
- Texas Instruments OMAP
- Analog Devices Blackfin® DSP
- Samsung S3C Application Processors
- Renesas SuperH

© 2007 QuickLogic Corporation

Chapter 3
Data Flow

This chapter describes the various ways in which data moves through an ArcticLink device. It contains the following sections:

- **"Functional Overview" on page 7**
- **"USB OTG Controller and Fabric Data Flow" on page 13**
- **"SD/SDIO/MMC/CE-ATA Controller and Fabric Data Flow" on page 18**
- **"Datapath and Usage Models" on page 20**

3.1 Functional Overview

The ArcticLink device is essentially a conduit for data. Using a combination of interfaces, controllers, and programmability, data is moved into the device, manipulated and then moved back out. This chapter focuses on the data movement and manipulation process. Data movement must be described in relation to the intended use of the device, for example, its use in a portable application product.

The data flows through these main components of the ArcticLink device:

- Hi-Speed USB 2.0 OTG Architecture
- SD/SDIO/MMC/CE-ATA Architecture
- Programmable Fabric
- Dual-Port Scratchpad 8 KB SRAM
- Fast Peripheral Bus Interface

© 2007 QuickLogic Corporation

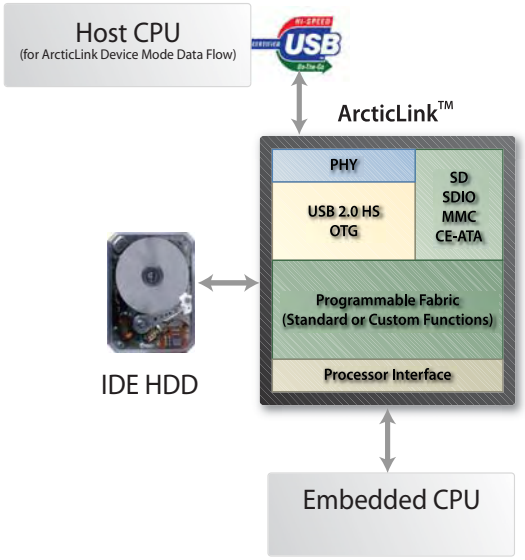
The flow of the data through the main components of the ArcticLink device are described as follows:

- Device Mode Data Flow—In device mode, the ArcticLink device is directed from another CPU, which acts as the "host" CPU and is connected via the USB port.

NOTE: Device Mode Data Flow functionality is only supported via the USB port of the USB OTG Controller.

Figure 3-1 is an example of device mode data flow, when an IDE HDD is connected.

Figure 3-1: ArcticLink's "Host" CPU Connectivity in Device Mode



© 2007 QuickLogic Corporation

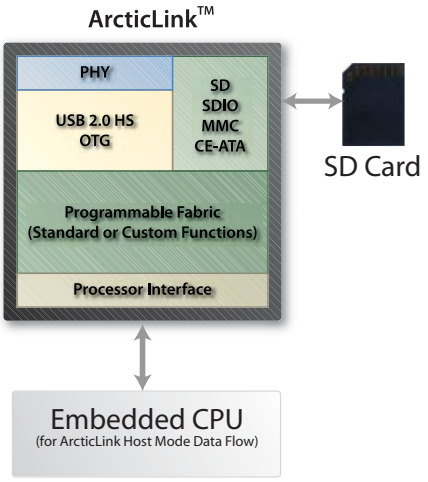
- Host Mode Data Flow—In host mode, the embedded CPU connected to the ArcticLink device via the programmable fabric is in control of the entire system and the internal scratchpad 8 KB SRAM. This 8 KB SRAM serves as an intermediary holding place used primarily for DMA transfers.

  However, the 8 KB SRAM must be set up in the embedded CPU's software to be the intermediary holding place for data in the SD/SDIO/MMC and CE–ATA Architecture.

NOTE: Host Mode Data Flow functionality can either be supported by the SD/SDIO/MMC/CE-ATA Host Controller port—or the USB OTG Controller USB port.

Figure 3-2 is an example of host mode data flow, when an SD card is connected.

Figure 3-2: ArcticLink's Embedded CPU Connectivity in Host Mode



In most cases an embedded processor would be required to handle transactions in both host and device modes. Other options do exist, depending on the complexity of operations and the modes to be handled i.e., a USB enumeration IP (available from QuickLogic) that can reside in the Programmable Fabric portion.

In the ArcticLink, the data flows along paths that pass through the main components. All possible combinations of active, or inactive, or both, device components are shown in **Table 3-1**, see below.

Table 3-1: System Operating Modes

USB OTG	SD/SDIO/MMC	Fabric	Supported?
-	-	Inactive	No
-	Device	Active	No
Inactive	Inactive	Active	Yes[a]
Host	Inactive	Active	Yes
Device	Inactive	Active	Yes
Inactive	Host	Active	Yes
Host	Host	Active	Yes
Device	Host	Active	Yes

a. Fabric design dataflow only

Table 3-2 lists the different ways that data can flow through the device. For a description of each operating mode listed below, see **"Datapath and Usage Models" on page 20**.

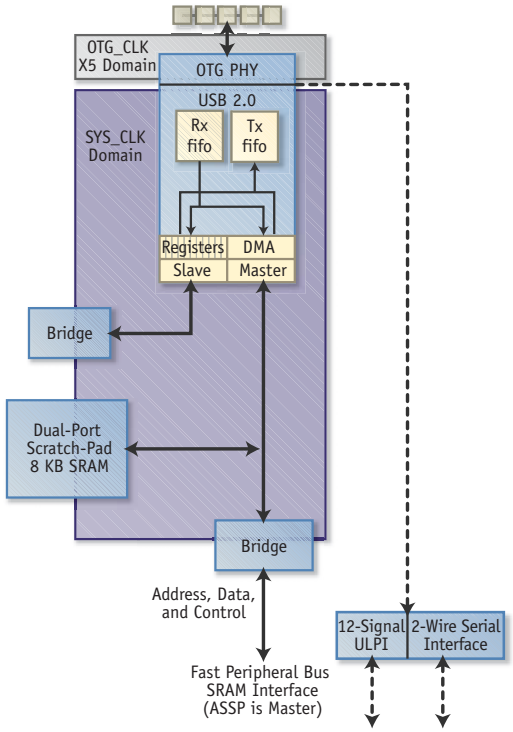
Table 3-2: Datapath Models

USB OTG Controller and Fabric					
Host			Device		
PIO Mode	USB DMA+ SRAM	USB DMA+ Bus Mastering	PIO Mode	USB DMA+ SRAM	USB DMA+ Bus Mastering
Yes	Yes	Yes	Yes	Yes	Yes
SD/SDIO/MMC and Fabric					
Host			Device		
PIO Mode	CPU DMA Mode		PIO Mode	CPU DMA Mode	
Yes	Yes		not supported	not supported	

3.1.1 Hi-Speed USB 2.0 OTG Architecture

A Hi-Speed USB 2.0 OTG controller with an on-chip scratchpad dual-port 8 KB SRAM memory are combined in the architecture.

Figure 3-3: Hi-Speed USB 2.0 Architecture



The main components are as follows:

- Hi-Speed USB 2.0 OTG Controller
- Fast Peripheral Bus (FPB)

© 2007 QuickLogic Corporation

The architecture provides Hi-Speed USB 2.0 OTG connectivity, and data buffering between the embedded CPU connected via the programmable fabric and the Hi-Speed USB 2.0 connection.

3.1.1.1 Hi-Speed USB 2.0 OTG Controller

The Hi-Speed USB 2.0 OTG Controller is a Dual-Role Device (DRD) controller that supports host and device functions. In host mode, the ArcticLink device's Hi-Speed USB 2.0 OTG controller core is responsible for retrieving data for movement to/from memory, and updating data on the Fast Peripheral Bus SRAM Interface. The on-chip dual-port scratchpad 8 KB SRAM buffer memory is the main source and sink for data in transit. In device mode, the application running on the host CPU connected via the USB port is responsible for initiating the transfer process for data pickup and storage.

The ArcticLink device's Hi-Speed USB 2.0 OTG controller core cannot operate in both modes simultaneously.

3.1.1.2 I^2C Compatible Serial Bus Interface

The ASSP/Fabric interface internal to ArcticLink contains a 2-wire serial interface that is compatible with the I^2C™ interface standard. This I^2C compatible serial bus interface can be used for carkit applications and FS OTG Transceiver control.

NOTE: These functions can be used if the ULPI (FS /LS) serial interface is selected, or carkit mode is selected—and not intended for other purposes (i.e., only a carkit PHY is supported in these modes).

The I^2C compatible serial bus interface can be used for control of a USB 1.1 Full Speed OTG Transceiver.

Please see QuickLogic's *ArcticLink Solution Platform Data Sheet* for more information about this interface.

3.1.2 SD/SDIO/MMC/CE-ATA Architecture

The SD/SDIO/MMC/CE-ATA Host Controller port provides a bidirectional interface to any 1-bit, 4-bit SD or SDIO peripheral component at speeds up to 50 MHz. The Rx and Tx FIFOs can be accessed from the programmable logic-based Peripheral Bus interface via a processor bus—or from a DMA engine optionally implemented within the programmable fabric.

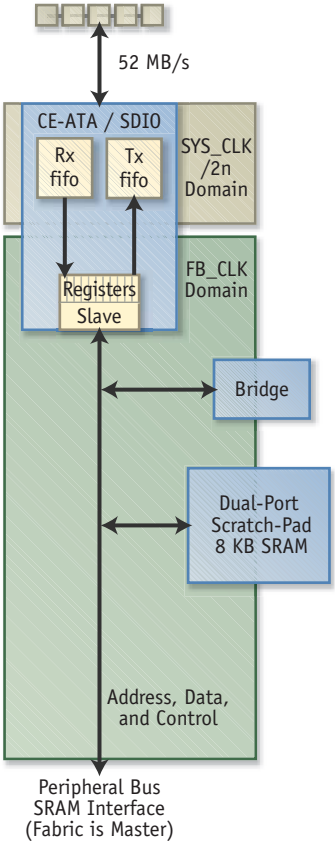
NOTE: A DMA engine used to access the FIFOs of the SD/SDIO/MMC/CE-ATA controller is an optional feature that can be implemented in the programmable fabric of the ArcticLink device.

In CE-ATA/MMC mode this interface provides a high-speed connection to any 1-bit, 4-bit or 8-bit CE-ATA or MMC compliant peripheral component at speeds up to 52 MHz.

The CE-ATA mode implements the CE-ATA command set including CMD-60 and CMD-61.The Rx and Tx FIFOs can be accessed from the programmable logic-based Peripheral Bus interface that allows the implementation of an DMA engine (optional) to manage SD, SDIO, MMC, or CE-ATA data traffic. This optional programmable logic based DMA implementation can manage traffic over the SD/SDIO/MMC/CE-ATA interface to minimize the utilization of the host processor for servicing data traffic to SD, SDIO, MMC, or CE-ATA peripherals.

© 2007 QuickLogic Corporation

Figure 3-4: SD/SDIO/MMC and CE-ATA Architecture



3.1.3 Progammable Fabric

The user-programmable fabric allows designers to implement their own custom functions into the ultra-low power ArcticLink programmable logic architecture which features:

- 20 Customizable Building Blocks (CBB)
- Seven blocks of dual-port SRAM. These seven blocks consist of one 8 Kb block and six 4 Kb blocks.
  - The 8 Kb block can be configured as 512x18 or 1024x9. The six 4 Kb blocks are 512x9 each. Pairs of adjacent 4 Kb blocks can be concatenated to increase effective data width or depth.
- Integrated FIFO controller logic
- Support for mobile DDR or SDR memory interfaces
- Configurable Clock Manager (CCM)
- Up to 120 GPIOs

Please see QuickLogic's *ArcticLink Solution Platform Data Sheet* for a description of the user-programmable fabric.

© 2007 QuickLogic Corporation

3.1.4 Dual-Port Scratchpad 8 KB SRAM

A scratchpad memory is provided on-chip for temporary storage as well as to facilitate data flow between the Hi-Speed USB 2.0 OTG controller and the external embedded processor connected via the programmable fabric.

The scratchpad is dual-ported with one port connected to the internal bus closely coupled with the Hi-Speed USB 2.0 OTG controller, while the other port is connected to the fabric mastered Peripheral Bus SRAM Interface and source clock. This enables the internal bus logic to run at a much higher clock frequency independent of the programmable fabric design frequency.

3.1.5 Fast Peripheral Bus Interface

Bus mastering is supported by most modern embedded processor memory bus architectures for the purpose of improving performance.

For systems where bus mastering is an option, the ArcticLink's on-chip Fast Peripheral Bus interface can support external bus mastering.

The Fast Peripheral Bus interface lets the internal DMA engine in the USB 2.0 OTG controller directly access System Memory through an on-chip memory controller optionally implemented in the programmable fabric. This direct access can be done without being under the control of a CPU.

NOTE: The programmable fabric memory controller is an optional feature that can be implemented in the ArcticLink device.

In addition, the Fast Peripheral Bus bridge manages a clock domain crossing to allow the internal bus clock to run independently of the programmable fabric memory controller clock.

3.2 USB OTG Controller and Fabric Data Flow

The flow of the data through the USB OTG architecture and Programmable Fabric component of the ArcticLink device are described as follows:

- Host Mode Data Flow
- Device Mode Data Flow

3.2.1 Host Mode Data Flow

When the ArcticLink device's USB controller is operating in host mode, its core is responsible for the pickup and/or storage of data in transit from the main 8K SRAM scratchpad memory.

Although the 8 KB on-chip scratchpad dual-port SRAM serves as the data source and sink, the data does not have to be created in the 8 KB SRAM memory when it is the source. Data can be created off-chip and then moved into the 8 KB SRAM by an embedded processor, in which case the 8 KB SRAM acts as a data sink. The data can then be moved to a second peripheral from the 8 KB SRAM, in which case the 8 KB SRAM acts as a source.

© 2007 QuickLogic Corporation

3.2.1.1 USB OTG Controller Example—Host Mode

In this example the ArcticLink device is acting as a Personal Digital Assistant DRD Controller in host mode.

Data for the Personal Digital Assistant DRD Controller can be generated off-chip by some other attached device such as a cell phone, see **Figure 3-5**. The data flows into the USB port of the Personal Digital Assistant DRD Controller which then transfers the data into the 8 KB SRAM. The data is in the form of a USB data packet. The software controlling the ArcticLink device examines the packet's header and determines the course of action.

Table 3-4 shows which datapath model is used for this example of host mode data flow.

Table 3-3: ArcticLink Host Mode—USB OTG Model

USB OTG	SD/SDIO/MMC	Fabric	Supported
Host	Inactive	Active	Yes

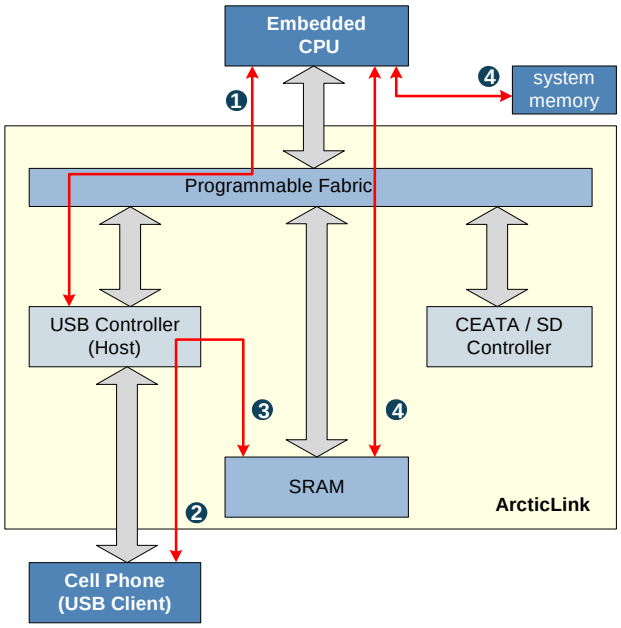
Table 3-4: Datapath Model —USB OTG Host

PIO Mode	USB DMA+ SRAM	USB DMA+ Bus Mastering
No	Yes	No

© 2007 QuickLogic Corporation

NOTE: The ArcticLink device and CPU are components in the PDA.

Figure 3-5: Personal Digital Assistant Host Controller Block Diagram



1(1)—PDA's CPU programs the USB Controller to receive**(send)** a data packet
-CPU sets up transfers via USB core by initializing channels, endpoints, and DMA descriptors

2(4)—Data on Cell Phone transfers to USB RxFIFO**(TxFIFO)**

3(3)—USB Controller's DMA engine writes**(reads)** data to**(from)** the SRAM and drains**(fills)** the RxFIFO**(TxFIFO)** in the USB core.
-USB core packetizes data to be transferred over the USB links
-ArcticLink device returns interrupt to PDA's CPU when transfer is done

4(2)—PDA's CPU retrieves**(sends)** data from**(to)** the SRAM with the PDA's CPU DMA engine and decides what to do with it
-For example CPU DMA engine can write**(read)** this to**(from)** system memory

NOTE: To send a packet follow the numbered steps in brackets (1), (2)...(4) and the direction shown in the brackets i.e., **(1)**—PDA's CPU programs the USB Controller to **(send)** a data packet.See **Figure 3-5**.

As illustrated by the example above, the PDA's CPU acts as a *traffic director*—it controls the flow of data around the ArcticLink device based on what it intends to do with the data.

3.2.2 Device Mode Data Flow

When the ArcticLink device's USB controller is operating in device mode, data transactions are triggered when the Host Computer sends either data or control packets through the USB link.

3.2.2.1 USB OTG Controller Example—Device Mode

In this example of device mode data flow, the ArcticLink device and CPU are components in the PDA which connect to a Host Computer, see **Table 3-5** and **Figure 3-6**.

Upon receiving a read control packet from the Host Computer, the USB controller on the ArcticLink device issues an interrupt to the embedded CPU in the PDA and tells it to load data into the 8 KB on-chip scratchpad SRAM and setup the DMA engine in the USB Controller.

The DMA engine in the USB controller then transfers the data from the 8 KB on-chip scratchpad SRAM into the USB controller's Tx FIFO for delivery over the USB link to the Host Computer.

Table 3-5: ArcticLink Device Mode—USB OTG Model

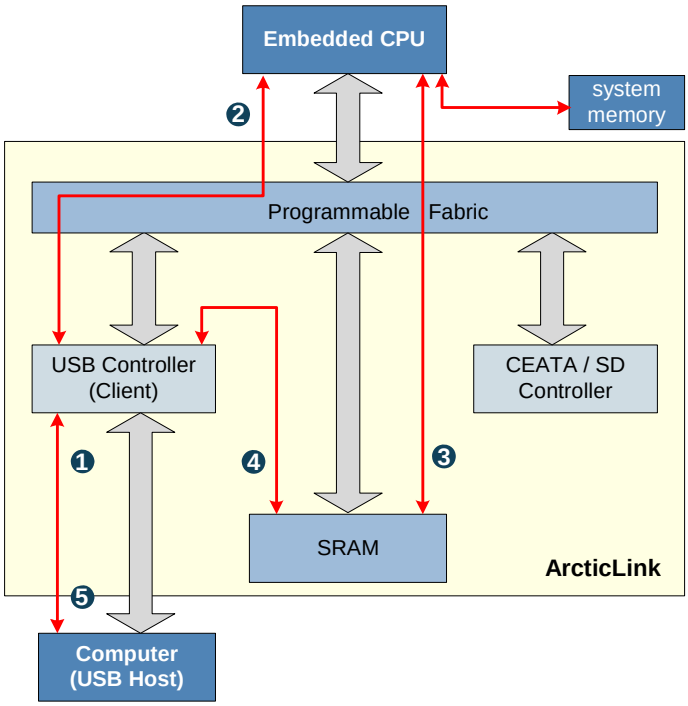
USB OTG	SD/SDIO/MMC	Fabric	Supported
Device	Inactive	Active	Yes

Table 3-6: Datapath Model—USB OTG Device

PIO Mode	USB DMA+SRAM	USB DMA+Bus Mastering
No	Yes	No

© 2007 QuickLogic Corporation

Figure 3-6: Personal Digital Assistant Device Controller Block Diagram



1(1)—USB Host's CPU sends the USB Controller a read(write) control packet through the USB link

2(4)—ArcticLink's USB Controller interrupts the PDA's CPU to load data to the SRAM and setup the USB Controller's DMA engine

-PDA CPU sets up transfers through the USB core's port by initializing channels, endpoints, and DMA descriptors

3(5)—PDA's CPU sends(retrieves) data to(from) SRAM with PDA's CPU DMA engine

4(3)—USB Controller's DMA engine reads(writes) data from(to) SRAM and fills(drains) the USB TxFIFO(RxFIFO)

-USB core packetizes data to(from) the USB Host before it is sent over the USB links

5(2)—USB Controller's TxFIFO(RxFIFO) delivers packet over USB link to the USB Host

-ArcticLink device returns interrupt to PDA's CPU when transfer is done

NOTE: To send a packet follow the numbered steps in brackets (1), (2)...(5) and the direction shown in brackets i.e., **(1)**—USB Host's CPU sends the USB Controller a **(write)** control packet through the USB link. See **Figure 3-6**.

© 2007 QuickLogic Corporation

3.3 SD/SDIO/MMC/CE-ATA Controller and Fabric Data Flow

The flow of the data through the SD/SDIO/MMC/CE-ATA architecture and Programmable Fabric component of the ArcticLink device are described as follows:

- Host Mode Data Flow
- Device Mode Data Flow—The flow of the data through the ArcticLink device's SD/SDIO Controller is not supported when it is operating in device mode, see **Table 3-7**.

Table 3-7: ArcticLink Device Mode—SD/SDIO/MMC/CE-ATA Model

USB OTG	SD/SDIO/MMC	Fabric	Supported
-	Device	Active	No

NOTE: Device Mode Data Flow functionality is only supported via the USB port of the USB OTG Controller.

3.3.1 Host Mode Data Flow

As described in **Section 3.2.1**, when operating in host mode the 8 KB on-chip scratchpad dual-port SRAM serves as the main source and sink for data in-transit. This 8 KB SRAM must also be programmed in the software to be the main storage for data in-transit to and from the SD/SDIO/MMC/CE-ATA controller.

For this SD/SDIO/MMC/CE-ATA controller example, suppose that the ArcticLink device is used as a host to interface to a WiFi card. Inserting a SDIO WiFi card into the card slot alerts the SD/SDIO/CE-ATA controller to the presence of the SDIO WiFi card.The ArcticLink device's SD/SDIO/CE-ATA controller host mode data transactions are triggered when the embedded CPU sends either data or control commands through the SDIO/SD/CE-ATA interface, see **Table 3-8** and **Figure 3-7**.

Table 3-8: ArcticLink Host Mode Dataflow—SD/SDIO/MMC/CE-ATA Model

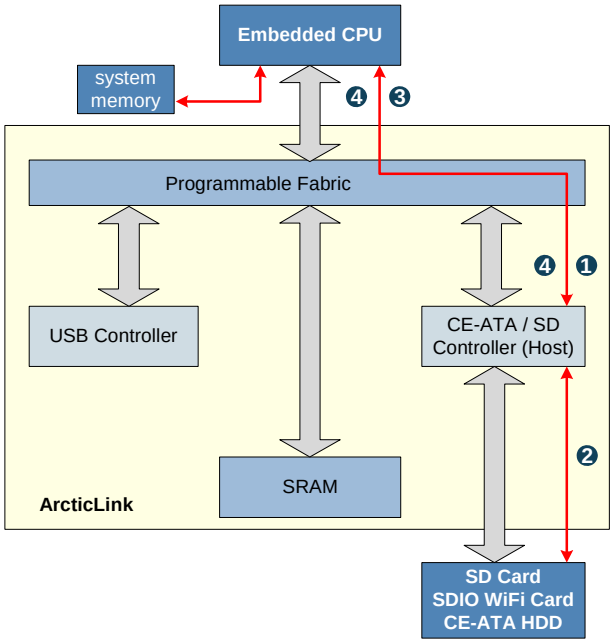
USB OTG	SD/SDIO/MMC	Fabric	Supported
Inactive	Host	Active	Yes

Table 3-9: Datapath Model—SD/SDIO/MMC/CE-ATA Host

PIO Mode	CPU DMA Mode
Yes	Yes

© 2007 QuickLogic Corporation

Figure 3-7: PDA and SDIO WiFi Card Dataflow Block Diagram



1—Embedded CPU programs CE-ATA/SD Controller to send a command
- **-**Initialize command parameters (CPU initializes host register)
- **-**Write command index (CPU writes to host register)
- **-**Request to send a command (host register requests to send command to CMD control)

1a—CE-ATA/SD Controller sends a command to SDIO WiFi card (or SD card, HDD)
- **-**Send index and argument (host register sends command via CMD control to SDIO WiFi card)

1b—SDIO WiFi Card sends a response to CE-ATA/SD Controller
- **-**Send a reply index and argument (SDIO WiFi card responds to CMD control with index and argument. CMD control forwards index and argument to host registers)

1c—CE-ATA/SD Controller sends a command completed response to CPU
- **-**Send a Command Complete Interrupt (host register interrupts CPU)
- **-**Read the response data (CPU reads response from host registers)

2—CE-ATA/SD Controller writes(reads) a block of data to(from) SDIO WiFi Card
- **-**Request a Data Interrupt (CE-ATA/SD controller sends interrupt to embedded CPU to request data)
- **-**Write a block of data (CPU writes block of data to the Data FIFO)
- **-**Send a datablock n data (Data FIFO sends a data block to the SDIO WiFi card via DAT control)
- **-**Send a DAT done response (DAT control sends a DAT done response to host registers)

3—CE-ATA/SD Controller sends a Transfer Complete Interrupt (host registers send a transfer complete interrupt to CPU) when finished

4—Embedded CPU reads(writes) data from(to) RxFIFO(TxFIFO) in **PIO mode** or with **CPU DMA engine** (i.e., into system memory)

NOTE: In this example of host mode data flow, the ArcticLink device and CPU are components in a PDA that is connected to a SDIO WiFi Card—or an SD card or a CE-ATA HDD.

© 2007 QuickLogic Corporation

3.4 Datapath and Usage Models

The following data flow modes are discussed:

- **“SDIO/SD/CE-ATA Datapath and Usage Model” on page 21**
- **“PIO (Slave) Mode Datapath Model” on page 22**
- **“Bus Mastering DMA Mode Datapath Model” on page 23**
- **“On-Chip (8 KB SRAM) Mode Datapath Model” on page 24**

© 2007 QuickLogic Corporation

3.4.1 SDIO/SD/CE-ATA Datapath and Usage Model

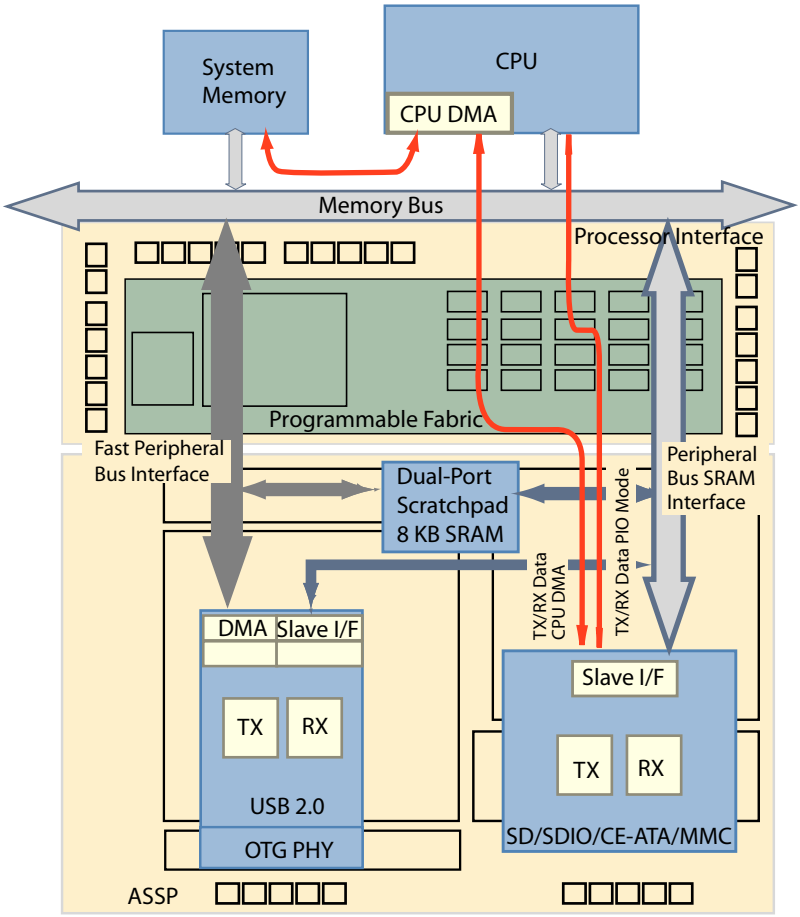
This SDIO/SD/CE-ATA datapath usage model focuses on the data flows and manipulations that occur when data is transferred across the SDIO/SD/CE-ATA controller, see **Figure 3-8**.

The following steps describe the datapath interactions of this model:

1. The CPU (connected via the Fabric interface) issues a read(write) command to the SD Host Controller.
2. Upon recieving a command, the SD Host Controller retrieves(writes) data from(to) the SDIO/SD/CE-ATA card (or HDD).
3. After it has finished reading(writing) the card (or HDD), the SD Host Controller returns an interrupt to the CPU.

NOTE: When the CPU reads(writes) data from(to) the FIFOs, it uses either the CPUs DMA engine or the data is transferred in PIO mode.

Figure 3-8: SDIO/SD/CE-ATA Datapath and Usage Model Block Diagram



3.4.2 USB Datapath Usage Models

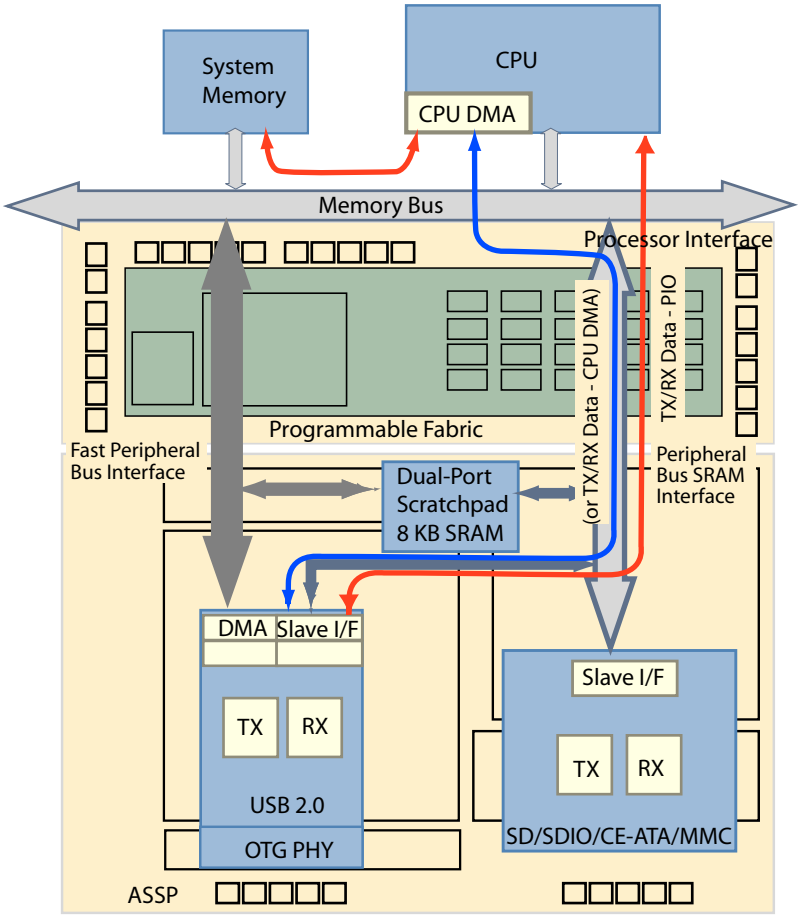
3.4.2.1 PIO (Slave) Mode Datapath Model

The USB PIO (Slave) datapath usage model gives a simple sequence of events that represent a typical datapath interaction, see **Figure 3-9**.

The following steps describe the datapath interactions in this mode:

1. The CPU (connected via the Fabric) sets up transfers (blue and red arrows) via the USB core's **slave port** by initializing channels and endpoints.
2. The CPU sends(retrieves) data to(from) the USBs internal FIFOs with the CPU DMA engine (blue arrow) or in PIO mode (red arrow).
3. When finished, the USB controller returns an interrupt (blue and red arrows) to the CPU.

Figure 3-9: PIO (Slave) Mode Block Diagram



© 2007 QuickLogic Corporation

3.4.2.2 Bus Mastering DMA Mode Datapath Model

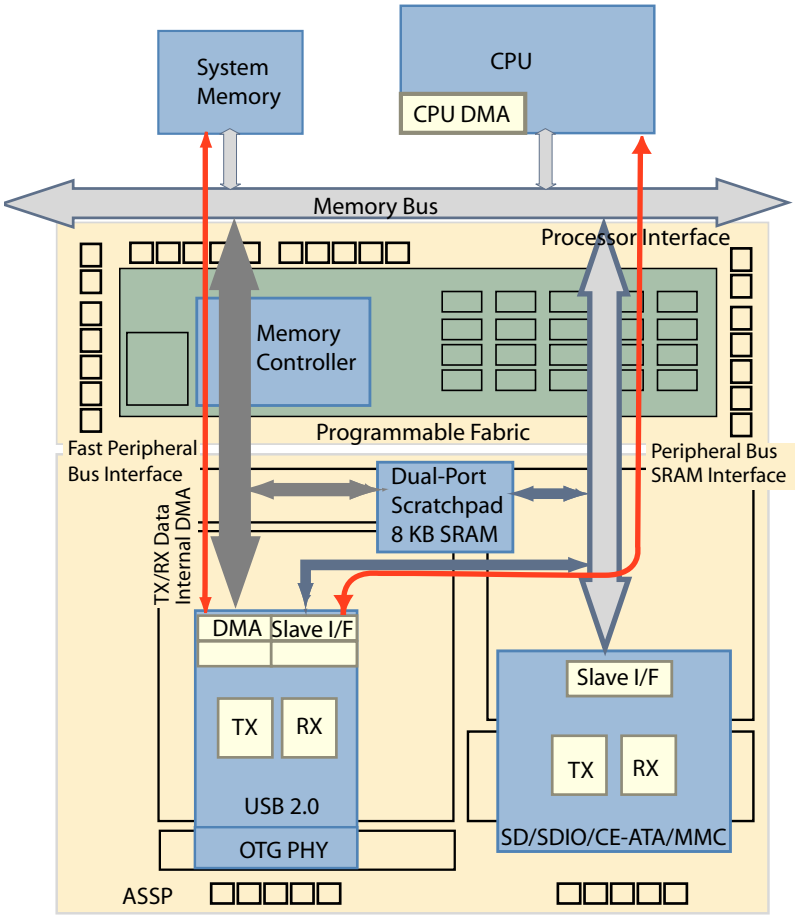
This mode refers to using the internal DMA engine in the ArcticLink's USB core with an SDRAM controller (optionally available) in the Programmable Fabric for bus mastering.

See the following steps for a description of the datapath interactions in this mode:

1. The CPU (connected via the Fabric) sets up the transfers via the USB core's **slave port** by initializing channels, endpoints, and DMA descriptors.
2. ArcticLink's USB core DMA engine reads(writes) data from(to) System Memory directly and fills(drains) the FIFO in the USB core.
3. The USB core packetizes the data to be transferred over the USB links.
4. ArcticLink returns an interrupt to the CPU when finished

NOTE: The advantage of this mode over non-bus mastering mode is the CPU does not need to setup multiple DMA engines, so less CPU and Bus bandwidth are used.

Figure 3-10: USB DMA with Bus Mastering Block Diagram



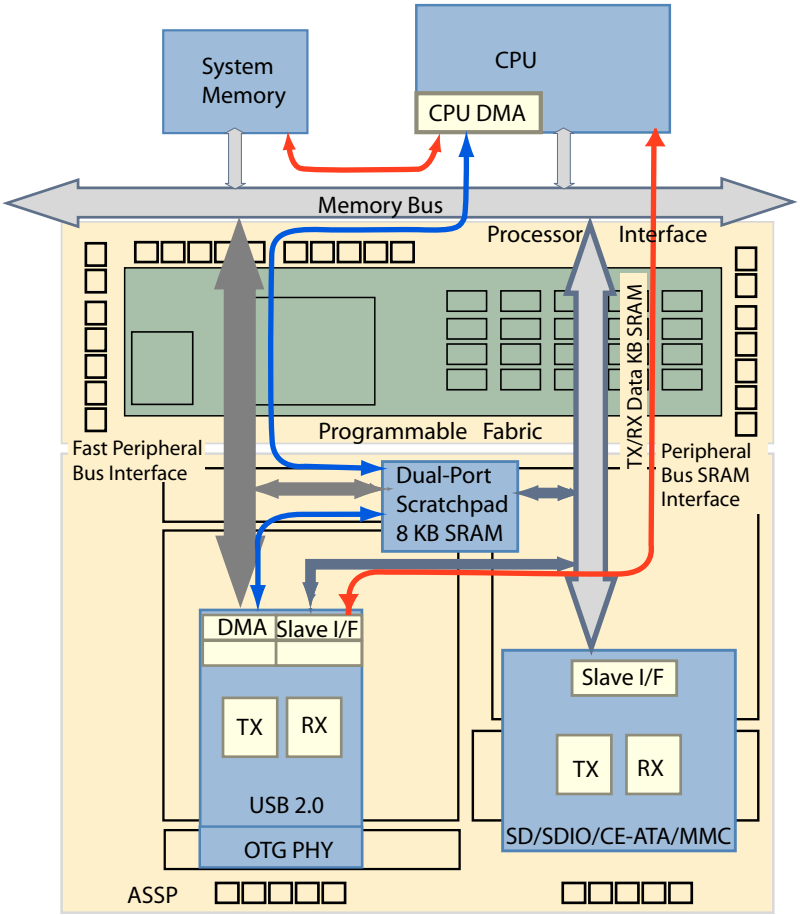
3.4.2.3 On-Chip (8 KB SRAM) Mode Datapath Model

This mode refers to using the internal DMA engine in ArcticLink's USB controller core, see **Figure 3-11**.

1. The CPU (connected via the Fabric) sets up transfers via the USB controller core's **slave port** by initializing channels, enpoints, and DMA descriptors (red arrow).
2. The CPU sends(retrieves) data to(from) the on-chip 8 KB SRAM with the CPU DMA engine.
3. ArcticLink's USB core DMA engine reads(writes) data from(to) the on-chip 8 KB SRAM (blue arrow) and fills(drains) the FIFO in the USB core.
4. The USB controller core packetizes the data to be transferred over links
5. ArcticLink returns an interrupt to the CPU when finished.
6. The CPU reads(write) data from(to) the on-chip 8 KB SRAM with the CPU DMA engine then writes(reads) to(from) System Memory.

NOTE: In this mode, less CPU utilization is required because the CPU does not have to take care of each block transfer. PIO mode does not share this advantage.

Figure 3-11: USB DMA Using the Internal 8 KB SRAM Block Diagram



© 2007 QuickLogic Corporation

Chapter 4
Hi-Speed USB 2.0 OTG Controller

This chapter provides a detailed description of the USB 2.0 OTG Controller functionality. It contains the following sections:

- **"Functional Overview" on page 25**
- **"Features" on page 27**
- **"Operation" on page 28**

4.1 Functional Overview

The Hi-Speed USB 2.0 OTG Controller contains the USB OTG Controller core, Integrated Phy, dedicated Rx and Tx FIFO, and dedicated DMA engine. The Hi-Speed USB 2.0 OTG Controller core connects to the peripheral bus SRAM interface (Fabric is master) or fast peripheral bus SRAM interface (ASSP is master) and communicates with the application and off-chip system memory. The Hi-Speed USB 2.0 OTG Controller is a Dual-Role Device (DRD) controller which supports host and device functions, and can be configured as Host-only or Device-only.

The Hi-Speed USB 2.0 OTG Controller can provide two modes of operation:

- Slave-Only mode (optional)
- Internal DMA mode
  - USB 2.0 OTG Controller core to/from scratchpad 8KB SRAM or Programmable Fabric

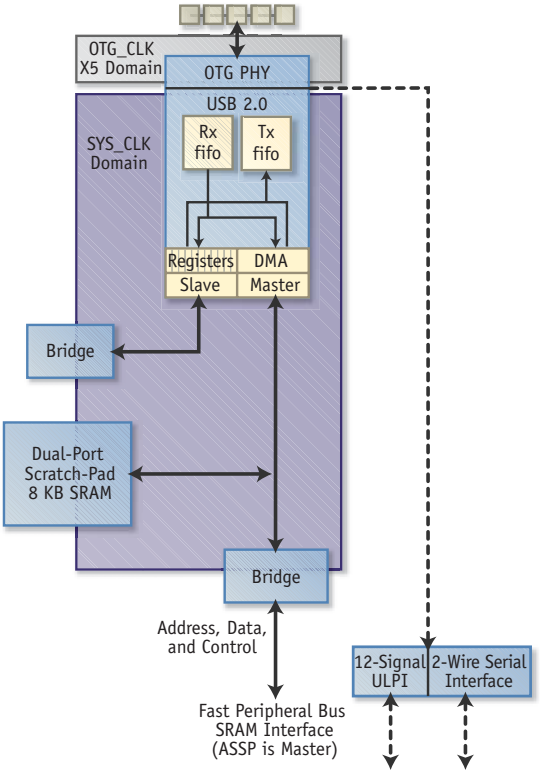
By default, Internal DMA mode is enabled. In this mode, the dedicated DMA controller manages the data transfer between System Memory (or the internal scratchpad 8KB SRAM) and the dedicated FIFOs through the fast peripheral bus SRAM interface. The fast peripheral bus SRAM interface allows the application to access the Rx and Tx Data FIFOs when Internal DMA is enabled, via the scratchpad 8 KB SRAM.

NOTE: With the available memory controller in the Programmable Fabric the internal DMA engine can do data transactions with off-chip System Memory.

© 2007 QuickLogic Corporation

Figure 4-1 shows a simplified block diagram of the Hi-Speed USB 2.0 OTG Controller.

Figure 4-1: Hi-Speed USB 2.0 OTG Controller Core Block Diagram



© 2007 QuickLogic Corporation

4.2 Features

The Hi-Speed USB 2.0 OTG Controller main features includes:

- The USB 2.0 OTG controller with Integrated PHY is compliant with the USB Revision 2.0 Specification.
- Integrated HS USB 2.0 OTG port capable of high speed (HS, 480 Mbps), full speed (FS, 12 Mbps), and low speed (LS, 1.5 Mbps) transfers
- Integrated PHY with dedicated Internal Phase-Locked Loop (PLL) with external 12 MHz input for low EMI
- Supports both Point-to-Point and Multi-Point (root hub) applications
- Optional ULPI HS/USB 1.1 FS Shared-Pin Interface via ASSP/programmable fabric interface
- Optional I2C compatible serial bus for OTG control in USB 1.1 FS mode available via ASSP/programmable fabric interface
- Double-buffering scheme for improved throughput and data transfer capabilities
- Supports OTG Host Negotiation Protocol (HNP) and Session Request Protocol (SRP)
- Supports suspend and remote wake-up
- Supports external charge pump source for applications requiring higher current requirements of VBUS
- Configurable power management features
- Integrated 5.2 KB memory
- Supports packet-based, dynamic FIFO memory allocation, for flexible, efficient use of RAM
- Total of sixteen endpoints comprising of:
  - One fixed bi-directional control endpoint
  - One software programmable IN or OUT endpoint
  - Seven IN endpoints
  - Seven OUT endpoints
- Optimization for the following applications and systems:
  - Portable electronic devices
  - Point-to-point applications (no hub, direct connection to HS, FS, or LS device)
  - Multi-point applications (as an embedded USB host) to devices (hub and split support)

© 2007 QuickLogic Corporation

4.3 Operation

4.3.1 USB 2.0 OTG Controller Initialization

NOTE: Skip this section and go to **4.3.1.1 External ULPI PHY**, if you are using an external ULPI PHY.

To gain access to the ArcticLink's USB 2.0 OTG controller requires writing to these register fields located in the common registers.

1. Set the USB_EN and the USB_OC_PHY_EN bits in the Enable Register (Offset Address: 0000010h).
2. Set the (OC_PHY_CK_EN) to enable the external 12 MHz USB PHY input clock and the system clock (SYS_CK_EN) bit in the System Clock Enable Register (Offset Address: 0000018h).
3. Set the USB_OTG_INT_EN bit in the Interrupt Enable Register (Offset Address: 000008h).

4.3.1.1 External ULPI PHY

Using an external ULPI PHY requires writing to these common registers to initialize the USB 2.0 OTG controller.

1. Set the SYS_CK_EN bit and clear the OC_PHY_CLK_EN bit in the System Clock Enable Register (Offset Address: 0000018h).
2. Set the USB_EN, GL_OC_PHY_SUSPEND, VBUS_COMP_DISABLE, and USB_OC_PHY_EN bits in the Enable Register (Offset Address: 0000010h).
3. Set the USB_OTG_INT_EN bit in the Interrupt Enable Register (Offset Address: 000008h).

4.3.2 Core Initialization

The application initializes the core from the configuration parameters in a specific sequence. The initialization sequence first reads the hardware configuration registers, and programs the applicable fields in the GAHBCFG and GUSBCFG registers, then writes to bits in the GINTMSK register and GUID register (if applicable). Lastly, the application reads the GINTS.CurMod bit. The GINTS.CurMod bit gives the operating mode, host or device.

Once the software has initialized the core, it must proceed through the Host mode and/or Device mode initialization steps.

The application sequentially initializes Host mode by programming the GINTMSK.PrtInt, HCFG and HPRT.PrtPwe registers then waits for the HPRT0.PrtConnDet interrupt bit. Next, the application programs the HPRT.PrtRst bit and waits for the HPRT.PrtEnChng interrupt before reading the HPRT.PRtSpd field. Lastly, it programs the HFIR, RXFSIZE, NPTXFSIZE,and HPTXFSIZ registers.

The application sequentially initializes Device mode by programming the DCFG, Device threshold control, and GINTMSK registers then waits for the GINTSTS.USBReset and GINTSTS.EnumerationDone interrupts.

© 2007 QuickLogic Corporation

See **"Register Descriptions" on page 86** for more information on these USB OTG registers.

4.3.3 Operating Modes

The application can operate the USB 2.0 OTG Controller core in internal DMA mode or Slave mode.

- In DMA mode, the application initiates data transfers between the scratchpad 8 KB SRAM and the USB 2.0 OTG Controller, and then yields control to the internal DMA engine to carry out the transfer. The DMA engine interrupts the application only on completion of transfers or error conditions.
- In Slave mode, the application is responsible for initiating and carrying out data transfers between System Memory and the USB 2.0 OTG Controller.

4.3.3.1 Internal DMA Mode

When the internal DMA option is chosen, the USB 2.0 OTG Controller core uses the fast peripheral bus SRAM interface to move data, transmit packet data fetch (fast peripheral bus SRAM interface to USB) and receive data update (USB to fast peripheral bus SRAM interface).

The DMA controller uses the programmed DMA address (HCDMAn register in Host mode and DIEPDMAn/DOEPDMAn register in Device mode) to access the data buffers.

There are two ways that the data is moved in this mode.

- Transfer-Level Operation
- Transaction-Level Operation

4.3.3.2 Slave Mode

When slave mode is chosen, the application can operate the USB 2.0 OTG Controller core in two ways to move data. In transaction-level (packet-level) operations, the application moves one data packet at a time for each channel/endpoint. In pipelined transaction-level operation, the application can program the USB 2.0 OTG Controller core to carry out multiple transactions.

- **Transaction-Level Operation —** The application performs transaction-level operations for a channel/endpoint for a transmission (host:OUT)/Device:IN), or a reception (host:(IN)/Device:OUT).
- **Pipelined Transaction-Level Operation —** The application pipelines multiple transactions back-to-back (IN or OUT).

4.3.4 Host Mode Operation in USB Transactions

The application must initialize one or more channels before it can communicate with connected devices.

Using the USB 2.0 OTG Controller as the host, the application must initialize a channel according to the type of USB transaction being performed.

© 2007 QuickLogic Corporation

Table 4-1 lists some types of USB transactions that are possible.

Table 4-1: USB Transactions

Transaction Type	Operation	Mode
Writing	Transmit FIFO	Slave
Reading	Receive FIFO	Slave
Bulk and Control	OUT/SETUP	Slave or DMA
Bulk and Control	IN	Slave or DMA
Control		Slave or DMA
Interrupt	OUT or IN	Slave or DMA
Isochronous	OUT or IN	Slave or DMA
Bulk and Control Split	OUT/SETUP or IN	Slave or DMA
Interrupt Split	OUT or IN	Slave or DMA
Isochronous Split	OUT or IN	Slave or DMA

4.3.4.1 USB Charge Pump

A charge pump (external) is necessary for the USB Host or the *USB OTG A–device* mode implementations to supply a +5 V source to attached devices.

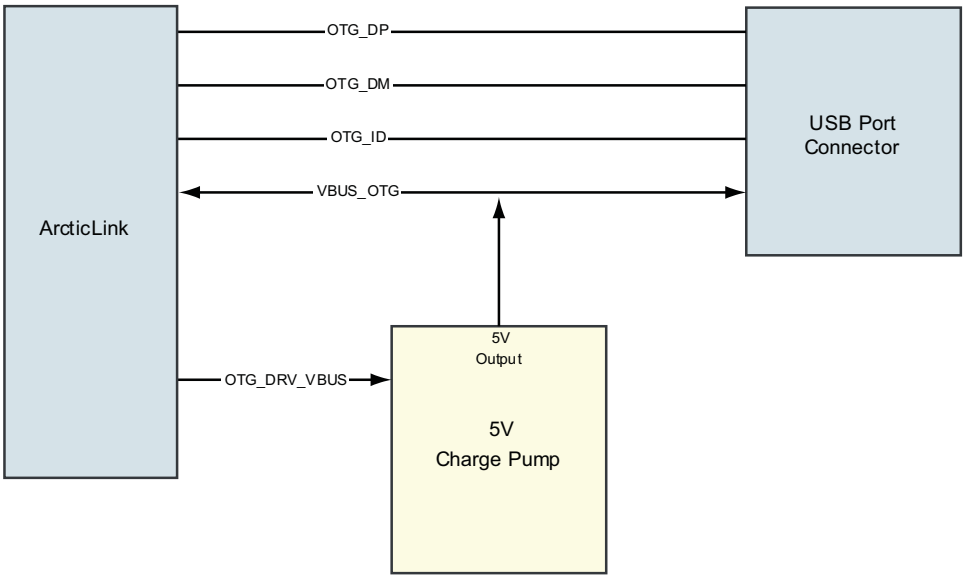
A large selection of charge pumps are available and should be selected by the designer depending on the system requirements such as the following:

- +4.8 V to +5.25 V *OTG–compatible* output on VBUS (i.e., VBUS_OTG)
- 8 mA (minimum for OTG devices), 100 mA (minimum if battery-powered), or 500 mA (recommended minimum) output current depending on the needs of devices to be attached to the charge pump.

Consult the selected charge pump data sheet for details on its connectivity.

Figure 4-2 illustrates a typical external charge pump implementation connected to the ArcticLink solutions platform and a USB connector.

Figure 4-2: USB Charge Pump Connectivity Block Diagram



© 2007 QuickLogic Corporation

4.3.5 Device Mode Operation In USB Transactions

The application must initialize one or more endpoints and set up the device core before it can handle transactions with connected devices

Table 4-2 lists the different endpoint initializations that are performed by the application depending on the specified transaction command received.

Table 4-2: Endpoint Initialization

Endpoint Initialization Required for
USB Reset
Enumeration Completion
SetAddress Command
SetConfiguration/SetInterface Command
Endpoint Activation/Deactivation
Device DMA/Slave Mode

Table 4-3 lists some types of USB transactions that are possible when the using the USB 2.0 OTG Controller in device mode.

Table 4-3: USB Transactions

Transaction Type	Operation
SETUP/OUT Data Transfers	Packet Read
	SETUP Transactions
	Global OUT NAK
	Disabling Out Endpoint
	Non-Isochronous OUT
	Isochronous OUT
	Incomplete OUT
	Stalling
Control Transfers	Write (SETUP, Data OUT, Status IN)
	Read (SETUP, Data IN, Status OUT)
	Two-Stage Transfer (SETUP/Status IN)

Table 4-4 lists the types of IN data transfers possible in device mode.

Table 4-4: IN Data Transfers

IN Data Transfer Operation
Non-Periodic Packet Write
Periodic Packet Write
Interrupt IN Endpoint
Setting Global Non-periodic IN Endpoint NAK
Setting IN Endpoint NAK
Disabling a Non-periodic IN Endpoint
Disabling a Periodic IN Endpoint
IN Endpoint Disable
Generic Non-periodic IN Data Transfers

© 2007 QuickLogic Corporation

Table 4-4: IN Data Transfers (Continued)

IN Data Transfer Operation
Generic Periodic IN Data Transfers
Incomplete Isochronous IN Data Transfers
Timeout for Non-periodic IN Data Transfers
Stalling Non-Isochronous IN Endpoints

4.3.6 Programming Models

NOTE: The terms IN and OUT are relative to the host. For example, during an IN transfer data flows from the device to the USB 2.0 Controller (host mode).

4.3.6.1 Host Mode DMA Bulk IN Transfer for Host OTG USB Core

Most Bulk IN USB transactions tend to be file transfers. This simple example describes how an application attempts to receive 1 maximum size data packet. The steps involved for this typical bulk IN operation in DMA mode, with the USB OTG core acting as a host, are described below.

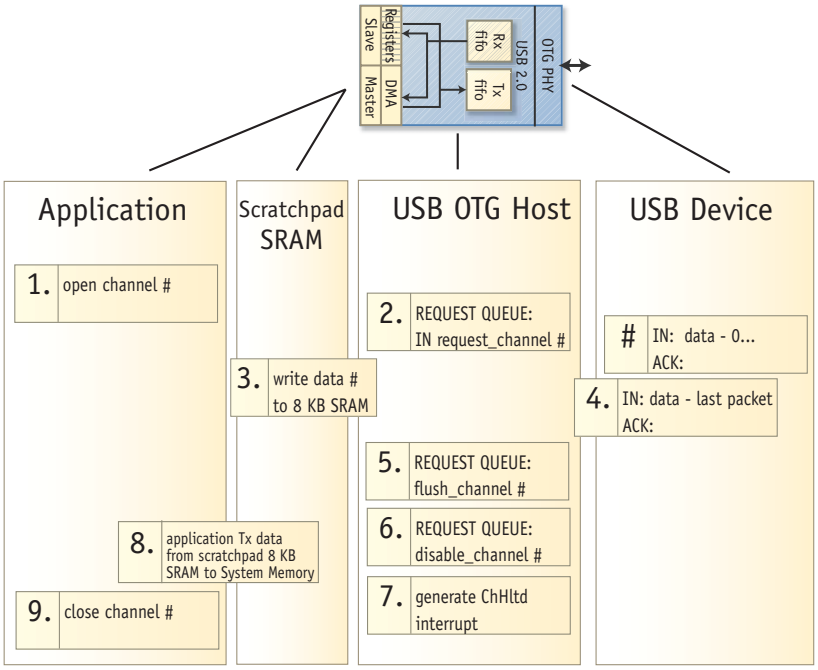
Table 4-5: Host Mode DMA Bulk IN USB Transactions

Transaction Type	Operation	Mode
Bulk and Control	IN	DMA

1. The application initializes and enables a channel #.
2. The USB OTG host writes an IN request to the Request queue when the channel # receives the grant from the arbiter.
3. The USB OTG host begins writing the received data to the dual-port scratchpad 8 KB SRAM (from here the processor can move the data to the system memory) immediately after the last byte is received with no errors.
4. After the last packet is received the USB OTG host sets an internal flag to remove any extra IN requests from the request queue.
5. The USB OTG host removes the extra requests.
6. A last request to disable channel # is written to the Request queue then channel # is internally masked for further arbitration.
7. The USB OTG host generates the ChHltd interrupt immediately when the disable request comes to the top of the queue.
8. Channel # is now available for other transfers in response to the ChHltd interrupt.

© 2007 QuickLogic Corporation

Figure 4-3: Host Mode DMA Bulk IN Transfer for Host OTG USB Core



4.3.6.2 Slave Mode Bulk IN Transfer for Device OTG USB Core

The steps involved for this typical bulk IN operation in slave mode, with the USB OTG core acting as a device, are described below.

Table 4-6: Slave Mode Bulk IN USB Transactions

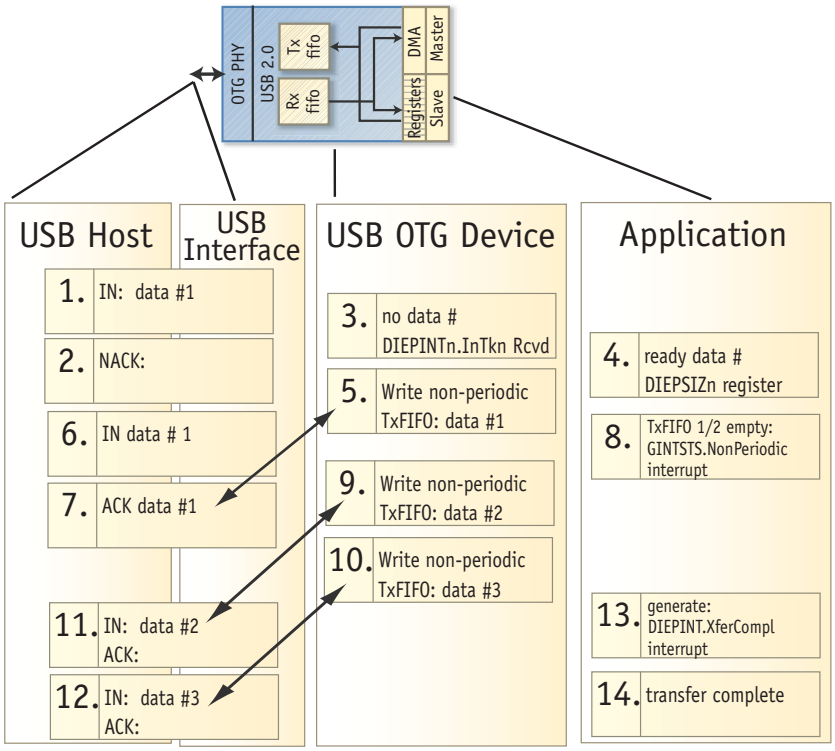
Transaction Type	Operation	Mode
Bulk and Control	IN	Slave

1. The host attempts to read data (IN token) from an endpoint.
2. When the IN token is received on the USB interface, the USB OTG core returns a NAK handshake, because no data is available in the transmit FIFO.
3. To inform the application was available to send, the USB OTG core generates an *DIEPINTn.InTkn Rcvd when TxFIFO Empty* interrupt.
4. When the data is ready, the application sets the DIEPTSIZn register with the transfer size and packet count.
5. The application writes one maximum transfer size or less of data to the Non-periodic TxFIFO.
6. The host re-attempts to read data (IN token).
7. Since the FIFO now has the data, the USB OTG core can respond to the host with the data afterwhich the host ACKs it.
8. As the TxFIFO becomes halfway empty, the USB OTG core generates a GINTSTS.NonPeriodic TxFIFO Empty interrupt. This will get the application to begin writing more data packets into the FIFO.

© 2007 QuickLogic Corporation

9. A data packet for the second transaction is ready in the TxFIFO.
10. A data packet for the third transaction is ready in the TxFIFO while the data for the second packet is sent to the USB.
11. The second packet is sent to the host.
12. Data transactions continue until the last packet is sent to the host.
13. The transfer is complete when the last packet is sent and the XferSize is zero.
14. The application processes the interrupt and determines if the transfer is complete from the setting of the DIEPINTn.XferCompl interrupt bit.

Figure 4-4: Slave Mode Bulk IN Transfer for Device OTG USB Core



© 2007 QuickLogic Corporation

Chapter 5
SD/SDIO/MMC/CE-ATA Controller

The SD/SDIO/MMC/CE-ATA controller provides a bridge between an external host processor (i.e. Marvell PXA) and an SD/SDIO/MMC or CE-ATA device.

This chapter contains the following sections to describe the SD/SDIO/MMC/CE-ATA controller.

- **"Functional Overview" on page 35**
- **"Features" on page 35**
- **"SD/SDIO/MMC/CE-ATA Host Controller" on page 36**

5.1 Functional Overview

The SD/SDIO/MMC/CE-ATA controller consists of the following two sub-modules:

- SD/SDIO/MMC/CE-ATA Host Controller. See "SD/SDIO/MMC/CE-ATA Host Controller" on page 36.
- SDIO System Interface

The SD/SDIO/MMC/CE-ATA controller can provide three modes of operation:

- SDIO mode
- CE-ATA mode
- MMC mode

5.1.1 SDIO System Interface

The SDIO system interface consists of the SD/SDIO/MMC/CE-ATA controller sub-module and the system interface logic.

The system interface logic connects the Programmable fabric SRAM interface to the Host Register Interface of the SD/SDIO/MMC/CE-ATA controller sub-module.

5.2 Features

The SD/SDIO/MMC/CE-ATA Controller's SDIO mode is compliant with *SD Host Controller Standard Specification, Version 2.0* and supports MMC Specification, Version 4.1. The SD/SDIO/MMC/CE-ATA Controller's CE-ATA mode is compliant with the *CE-ATA Digital Protocol, Revision 1.1RC* and based on the *SD Host Controller Standard Specification, Version 2.0*. The SD/SDIO/MMC/CE-ATA controller's MMC mode supports MMC Specification 4.1 with the exception of SPI mode.

© 2007 QuickLogic Corporation

The main features in SDIO mode include:

- Dynamic buffer management to increase data throughput. See “Dynamic Buffer Management” on page 38.
- Available DREQ signal for Marvell PXA processor to improve processor DMA performance

The following main features are supported:

- up to 52 MHz clock rate
- SD and SDIO 1-bit and 4-bit data mode
- I/O commands 52 and 53
- MMC in 1-bit, 4-bit, and 8-bit modes
- SDIO device interrupt in 1-bit and 4-bit modes
- Block size up to 512 bytes

The main features in CE-ATA mode include:

- Dynamic Buffer Management to increase data throughput. See “Dynamic Buffer Management” on page 38.
- Available DREQ signal for Marvell PXA processor to improve processor DMA performance

The following main features are supported:

- up to 52 MHz clock rate
- Clock rate up to 52 MHz
- CE-ATA commands 39, 60 and 61
- CE-ATA Command Completion Interrupt from device to host
- CE-ATA Command Completion Signal Disable protocol generation
- Block size up to 512 bytes

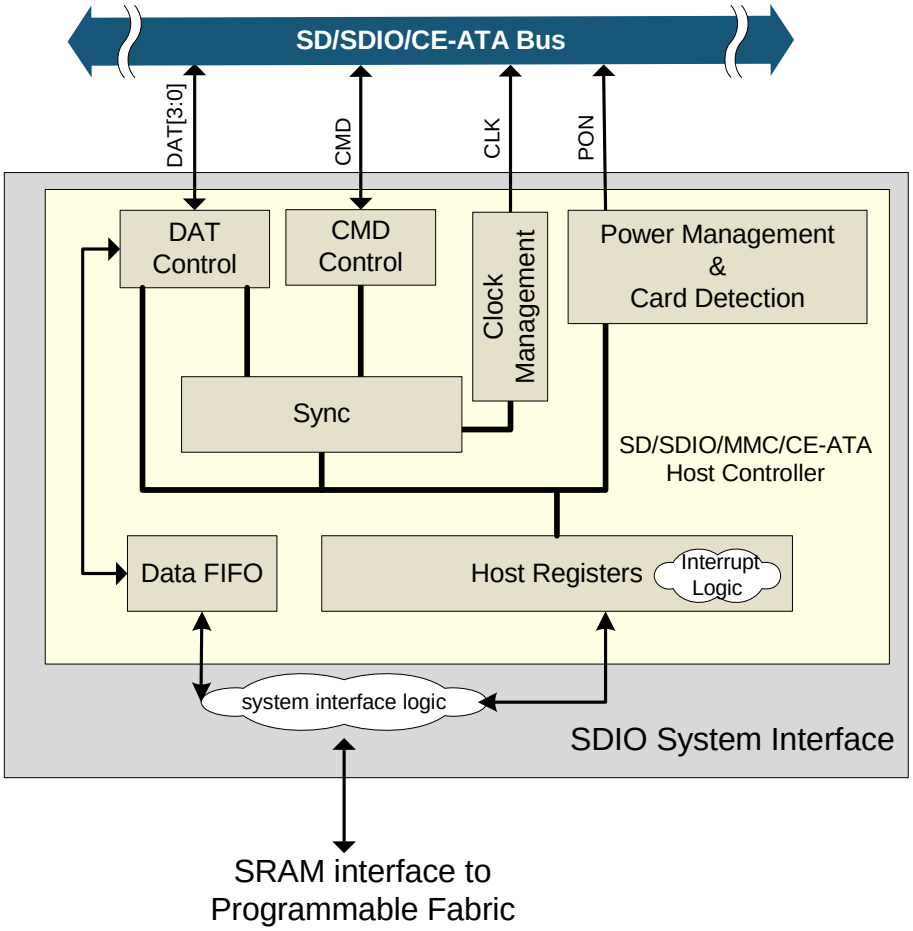
5.3 SD/SDIO/MMC/CE-ATA Host Controller

The SD/SDIO/MMC/CE-ATA Host Controller sub-module provides a bridge between a processor and either an SD/SDIO/MMC card or CE-ATA device and consists of the following components:

- Host Registers
- Data FIFO
- CMD Control
- DAT Control
- Clock Management
- Power Management and Card Detection
- Synchronization Block

© 2007 QuickLogic Corporation

Figure 5-1: SD/SDIO/MMC/CE-ATA Host Controller Block Diagram



5.3.1 Host Registers

The Host Registers block consists of the following components:

- Standard SD/SDIO host register set with an additional register for CE-ATA support, see **"CE-ATA Control Register (Offset 100h)" on page 77**
- Interrupt logic
- Auto CMD12 logic

The standard SD/SDIO host register set consists of seven parts:

- SD command generation—parameters to generate SD commands
- Response—response value from the card
- Data port—data access port to the internal buffers
- Host controls—host mode control and status
- Interrupt controls—interrupt statuses and enables
- Capabilities—Host Controller capability information

© 2007 QuickLogic Corporation

– Common area—common information area

The Host Controller provides a 32-bit system bus interface and supports byte accesses to the Host Registers. If the system bus is 8 or 16 bits wide, the system interface logic in the SDIO system interface converts it to a 32-bit width.

The Auto CMD12 Logic automatically issues a CMD12 command after the last block of data is transferred if the Auto CMD12 Enable bit in the Transfer Mode register is set.

5.3.2 Data FIFO

There are two asynchronous 512 byte Data FIFOs:

- TX_FIFO—for buffering transmit data
- RX_FIFO—for receiving data

The TX_FIFO provides the *Buffer Write Enable* flag to the Host Registers block. When the TX_FIFO is empty, *Buffer Write Enable* is asserted to indicate that the TX_FIFO is ready to accept the next block of data.

The RX_FIFO provides a *Buffer Read Enable* flag to the Host Registers block. When the RX_FIFO receives a block of data from the card, *Buffer Read Enable* is asserted to indicate to the processor that data is ready for pickup.

The *Buffer Read Enable* and *Buffer Write Enable* flags are available as read only bits in the Present State Register.

5.3.3 Dynamic Buffer Management

Dynamic Buffer Management is used to increase data throughput by enabling simultaneous two-way FIFO operation as described above. Setting the FIFO_EN bit(s) in the CE-ATA Control Register enables this feature.

If Dynamic Buffer Management is disabled, the FIFO operates sequentially at its RX_FIFO and TX_FIFO interfaces. First, the buffer has to be filled before its inbound data flow is suspended and the outbound data transfer operation can start. When the entire buffer has been transferred, the inbound data flow resumes.

When Dynamic Buffer Management is enabled the FIFO resumes filling as soon as there is room for at least 4 bytes of data to flow into the FIFO, after outbound transfer has been started. The FIFO can be dynamically filled and transferred at the same time, without waiting for it to completely finish off the previous cycle. This is applicable for both TX_FIFOs and RX_FIFOs.

5.3.4 CMD Control

CMD Control is responsible for sending commands, receiving responses, and checking errors. It contains a state machine and the Command Line module.

The Command Line module converts the command index and argument from the Host Registers block to serial format, and sends them to the CMD line. If there is a response from the card, the Command Line module also unpacks the serial format response packet into octets and stores them in the Host Registers block.

© 2007 QuickLogic Corporation

During the sending command and receiving response process, the Command Line generates a Cyclic Redundancy Code (CRC). It sends the CRC following a command argument, and checks the CRC after receiving the response data field.

The CMD Control state machine controls the Command Line module. There is a counter that works with the state machine to specify timing for the SD command and response packet.

To send a command, the SD/SDIO/MMC/CE-ATA Host Controller driver writes a command index value into the Command Register. The Host Register block then sends a request signal to the CMD Control state machine.

Upon receiving a request from the Host Registers block, the CMD Control state machine:

1. Assembles the command packet and enables output. The Command Line module shifts out the command packet and generates the CRC.
2. After 48 bits are sent out, the CMD Control state machine goes back to IDLE if there is no response. The current sending command cycle is completed.
3. If a response is expected, the CMD Control state machine waits for the response packet.
4. If response does not come within 64 clock cycles, the CMD Control state machine goes to IDLE and issues a timeout error interrupt.
5. After receiving the first bit of the response packet, the CMD Control state machine starts to count the number of bits received. The Command Line module converts the incoming serial data into parallel data, checks the command index field, generates a CRC, and verifies the received CRC value.
6. Based on the response packet type, the CMD Control state machine ends response packet reception after either 48 or 136 bits.

The CMD Control state machine also provides command end and response end status to DAT Control so that it can start data transfer based on the transfer mode settings.

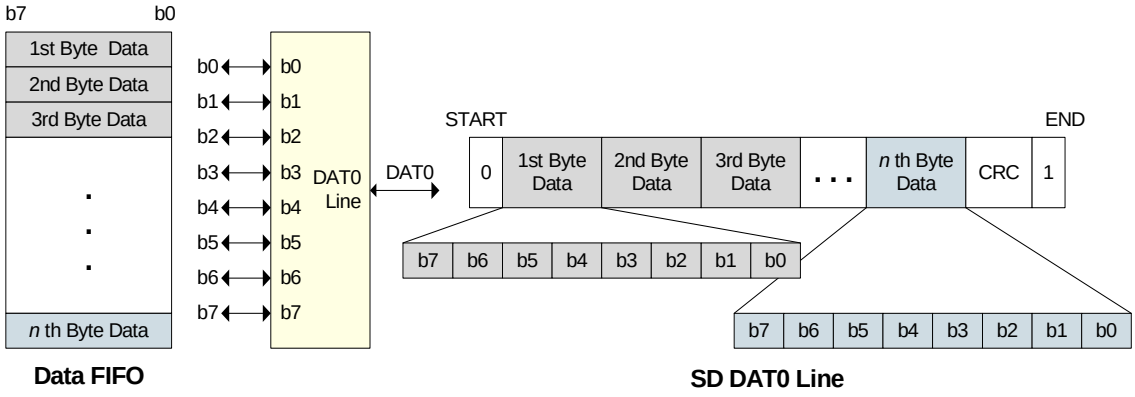
5.3.5 DAT Control

DAT Control is responsible for transferring data between the Data FIFO and the SD data bus, and for detecting the SDIO interrupt. It consists of an SD data multiplexer, a state machine, and DAT Line sub-modules.

The SD data multiplexer supports 1-bit and 4-bit SD/SDIO modes. In 1-bit mode, only DAT0 is used. All 8-bit data from the Data FIFO connects to the DAT0 Line sub-module in the following way: bit0 from the Data FIFO connects to bit0 of the DAT0 Line module, bit1 from the Data FIFO connects to bit1 of the DAT0 Line module, etc. The data bit sequence on the SD data line is as shown in **Figure 5-2**.

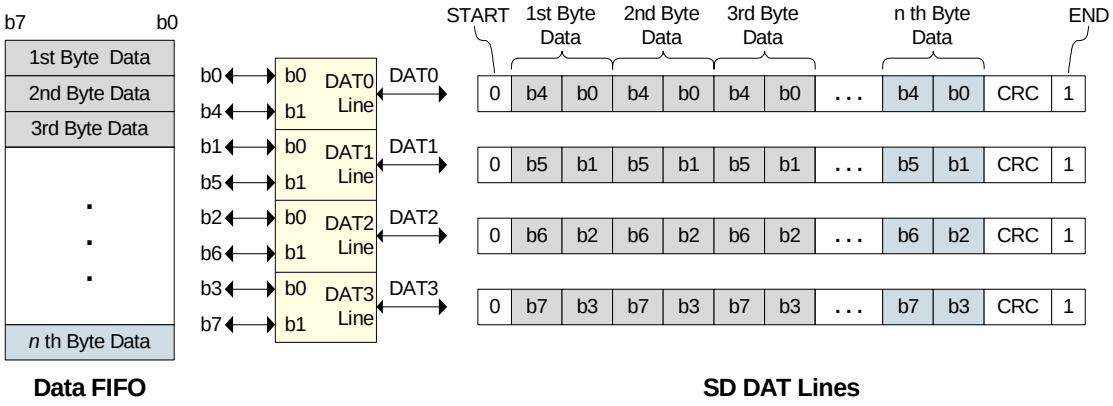
© 2007 QuickLogic Corporation

Figure 5-2: 1-bit SD Data Mode



In 4-bit mode, all four SD data lines DAT [3:0] are used. **Figure 5-3** shows data connections between the Data FIFO and four DAT Line modules in 4-bit mode. The data sequence in 4-bit mode is also shown.

Figure 5-3: 4-bit SD Data Mode



The DAT Control state machine instructs the DAT Line sub-modules to send data and generate the CRC, or to receive data and verify the CRC. It determines the end of the data packet based on the block length counter. It also detects timeout condition during busy state or waiting for read data.

Whenever the `stop` (or an `I/O abort`) command is issued, the DAT Control state machine goes to IDLE state immediately.

The SDIO interrupt function is implemented in the DAT Control module. In 1-bit SD mode, the DAT1 line is assigned as a dedicated interrupt pin. It is active low. In 4-bit mode, the SDIO interrupt logic detects the interrupt event on the DAT1 line during the interrupt period. For the definition of the interrupt period, refer to *SD Specifications, Part E1: SDIO Specification Version 2.0*.

© 2007 QuickLogic Corporation

5.3.6 Interrupt Control Logic

The SD/SDIO/MMC/CE-ATA host controller can serve a number of interrupt sources. Interrupt control logic consists of these register types: Interrupt Status, Interrupt Status Enable, and Interrupt Signal Enable. **Figure 5-4** shows the interrupt control logic which consists of these interrupt sources enabled as interrupts or system wakeup signals.

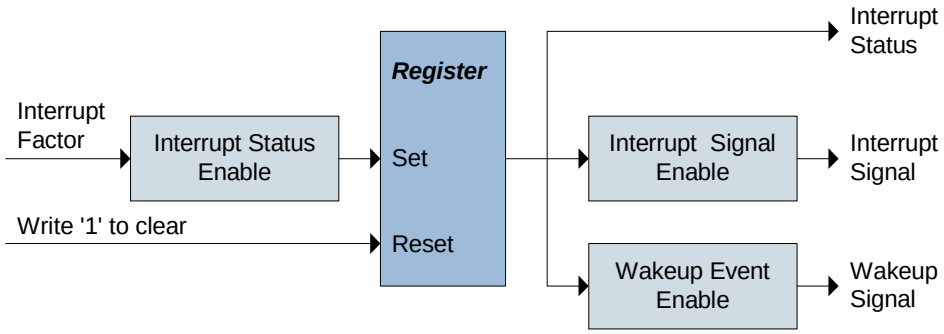
To be able to poll an interrupt event(s), the corresponding bit(s) must be unmasked in the Interrupt Status Enable register, i.e. “Normal Interrupt Status Enable Register (Offset 34h)” for the SD/SDIO/MMC/CE-ATA Host Controller.

As soon as the interrupt factor occurs the bit in the Interrupt Status register, i.e. “Normal Interrupt Status Register (Offset 30h)”, signals an interrupt event occurred by changing its bit value from 0 to 1. To clear the event and set the SD/SDIO/MMC/CE-ATA Host Controller to catch the next interrupt event, write 1 to the corresponding bit in the Interrupt Status register. The bit will then be cleared.

When the designer does not want to react on the interrupt factor anymore, the corresponding bit in the Interrupt Status Enable register, should be cleared so the interrupt is masked.

If the expected interrupt is to be a physical signal (i.e. not poll mode but interrupt mode), the corresponding bit has to be set in the Interrupt Signal Enable register also, i.e. “Normal Interrupt Signal Enable Register (Offset 38h)”.

Figure 5-4: Interrupt Control Logic



5.3.7 Clock Management

The Clock Management block is used to divide and enable/disable the SD clock. It consists of a Clock Divider and SD Clock Control Logic.

The Clock Divider is a loadable counter. When a new value is written to the Clock Control register, the Host Register block loads this new value to the Clock Divider. Whenever the internal clock enable bit in the Clock Control register is set, the Clock Divider starts to count down from the loaded value to ‘0’.

The SD Clock Control Logic controls SD clock toggling. When the Clock Divider reaches zero and SD clock enable bit in the Clock Control register is set, the SD clock toggles.

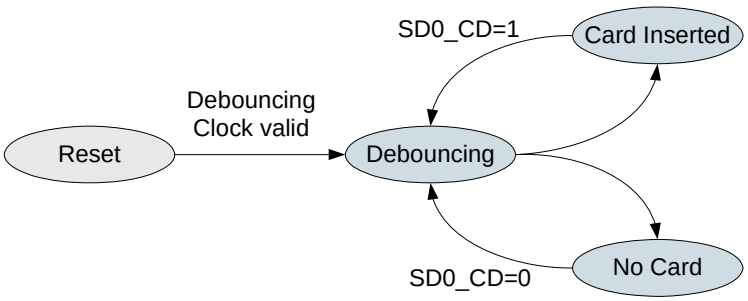
© 2007 QuickLogic Corporation

5.3.8 SDIO Power Management and Card Detection

The Power Management logic is designed for power saving purposes. It can turn SD bus power on or off by controlling the external power supply component. To turn the SD bus power on, '1' is written to the SD Bus Power bit in the Power Control register. To turn the SD bus power off, '0' is written to the SD Bus Power bit in the Power Control register.

The Card Detection logic detects SD card insertion and removal events. It consists of a state machine and a debouncing counter. **Figure 5-5** shows the Card Detection state machine.

Figure 5-5: Card Detection Finite State Machine



The state machine starts in **Reset** state at power-on and changes to the **Debouncing** state once the debouncing clock is valid. In the Debouncing state, provided that the SD0_CD signal is stable for more than 256 clocks, the state changes to a **Card Inserted** state if SD0_CD is '0' or to **No Card** if SD0_CD is '1'. If the card is removed, the state machine immediately moves from Card Inserted state to Debouncing state. In the same way, when the card is inserted, the state machine moves from No Card state to Debouncing state.

5.3.9 SD/SDIO/MMC/CE-ATA Host Controller Interface

5.3.9.1 Clocks and Reset

The SYS_RESET_n single function pin signal is used as a hardware reset for the SD/SDIO/MMC/CE-ATA Host Controller and all other ASSP blocks. Keep reset active (low) until the system clock (SYS_CLK) becomes stable. The system clock is not dedicated to the SD/SDIO/MMC/CE-ATA host controller, but is shared with other control modules.

NOTE: Reading or writing to the System Clock Enable Register (000018h) enables or disables the clocks to the individual controllers in the ArcticLink device.

FB_clk is used in modules such as registers and data FIFOs that interact with the host system bus. FB_clk is used to generate the peripheral (FPGA fabric) bus interface clock.

The system clock is used as the peripheral bus clock and is divided by 2n to generate the SD/SDIO/MMC/CE-ATA bus clock.

SD/SDIO/MMC/CE-ATA bus clock (i.e. SD Clock frequency) = Base Clock frequency/ divisor.

© 2007 QuickLogic Corporation

The divider value is set in the Clock Control Register (Offset 02Ch) as described by the SD Host Controller Standard Specifications and has a supported range of n= 1, 2, 4, 8, 16, 32, 64. The SD/SDIO/MMC/CE-ATA Host Controller driver uses the 2n value to calculate the SD Clock frequency. The Base Clock frequency for SD clock is set in the "Capability Register (Offset 40h)".

Table 5-1 summarizes the maximum frequency for each clock.

Table 5-1: Maximum Clock Frequency

Clock Signal Name	Maximum Frequency
SYS_CLK	111 MHz
FB_clk	111 MHz

5.3.9.2 SD/SDIO Signals

AF_VOLT_SUP is fed from the programmable fabric to the SD/SDIO/MMC/CE-ATA Host Controller in the ASSP. This signal is used to select the voltage supported by the ArcticLink SD/SDIO/MMC/CE-ATA Host Controller and the SD/SDIO plug-in card. The voltage supported by the Host Controller is stored in the SD/SDIO/MMC/CE-ATA Host Controller Capability register. The voltage supported by the SD/SDIO plug-in card is stored in the card's OCR register.

SD0_Pwr_on is used to turn the SD/SDIO plug-in card's power supply on or off.

The SD0_Pwr_on, SD0_LED_on, SD0_CD, and SD0_WP signals are standard SD/SDIO bus signals.

ASSP_SD0_DATA[0:3], ASSP_SD0_CLK, ASSP_SD0_CMD are multi-function pins that provide standard SD/SDIO bus signals when AF_SD0_EN=1. If AF_SD0_EN=1 is not true then these pins are normally GPIO pins.

For more information, refer to the *SD Specifications - Part1: PHYSICAL LAYER Specification*.

5.3.10 CE-ATA Initialization Sequence

5.3.10.1 CE-ATA Controller Initialization

To gain access to the ArcticLink's CE-ATA controller requires the initialization of these register fields in the Common registers set.

1. Set the CE_ATA_CK_EN bit(s) in the System Clock Enable Register. See 'System Clock Enable Register (Offset Address: 0000018h)" on page 72.
2. Set the CE_ATA_EN bit(s) in the Enable Register. See 'Enable Register (Offset Address: 0000010h)" on page 70.
3. Set the CE_ATA_INT_EN bit(s) in the Interrupt Enable Register. See 'Interrupt Enable Register (Offset Address: 000008h)" on page 69.

After this the register fields listed below in "CE-ATA IP Core Software Reset" are enabled.

© 2007 QuickLogic Corporation

5.3.10.2 CE-ATA IP Core Software Reset

Every time after the CE-ATA core is reset by Software, the following register fields should be set in the CE-ATA Control Register. The CE-ATA Control Register is part of the SDIO/CE-ATA Host Controller Register set, see 'CE-ATA Control Register (Offset 100h)" on page 77.

1. Set the FIFO_EN bit(s).
2. Set the CEATA_MODE_EN bit(s).
3. Set the CEATA_CCS_INT_EN bit(s).
4. (DMA only) If DMA is used then set the DREQ_MODE bit(s).

© 2007 QuickLogic Corporation

Chapter 6
Programmable Fabric

This chapter describes the user-programmable fabric of the ArcticLink device. More complete information on the Programmable Fabric, including timing data, is available in the *QuickLogic ArcticLink Solution Platform Data Sheet*.

This chapter contains the following sections:

- **"General Programmable Fabric Features" on page 45**
- **"ASSP/Programmable Fabric Interface Ports" on page 46**
- **"Designing with ArcticLink in QuickWorks" on page 47**
- **"Simulation Modeling" on page 51**

6.1 General Programmable Fabric Features

Table 6-1 and **Table 6-2** show the ArcticLink Fabric Features.

Table 6-1: ArcticLink Fabric Features

Features	QL1A100
Maximum Programmable Logic Gates	100,000
Logic Cells	640
Max I/O	120
RAM Modules	7
FIFO Controllers	7
RAM bits	36,864
CCM	1

Table 6-2: QL1A100 Maximum Usable I/Os

Device	VCCIO Banks				Total Maximum Usable I/Os
	Bank A	Bank B	Bank C	Bank D	
121 TFBGA (8mm x 8mm)	10	20	12	18	60
196 TFBGA (12mm x 12mm	30	34	12	44	120

© 2007 QuickLogic Corporation

6.2 ASSP/Programmable Fabric Interface Ports

The following **Table 6-3** outlines the different Anti-fuse categories. Please see QuickLogic's *ArcticLink Solution Platform Data Sheet* for detailed descriptions on these bits.

Table 6-3: Anti-Fuse Signal Categories

Category	Anti-fuse Signals	Description	Usage
System	AF_QL_REV [3:0]	These bits are primarily for the OS to identify the revision of the device. They are intended for use by the QL companion device designers to define the version of the IP programmed in the Fabric.	Set by the Fabric IP designer.
	AF_QL_ID [7:0]	These bits are primarily for the OS to identify the ID of the device. They are intended for use by the QL companion device designers to define the ID of the IP programmed in the Fabric.	Set by the Fabric IP designer.
	AF_SDIO0_CFG AF_SDIO1_CFG AF_PCI_CFG AF_IDE_CFG	These bits are primarily for the OS to identify the features programmed into Fabric of the ArcticLink device. The OS may use this information to load the appropriate S/W drivers for the appropriate device configuration.	Set to indicate the features available in the ArcticLink device. 1 = Feature present 0 = Feature not available
	AF_SRAM_MAP AF_BUS_WIDTH	These bits indicate the configuration of the host CPU (i.e. bus width and address decoding)	Set to the appropriate values if a host CPU is present.
USB OTG	AF_FB_VL[3:0] AF_ULPI_EXT_EN AF_USB_CFG	These bits are used to configure the USB OTG core in the ASSP.	Set to the appropriate values if the USB OTG is to be utilized.
SD/SDIO/CE_ATA	AF_ATA_CFG AF_SDCLK_FREQ AF_TIMEOUT_UNIT AF_TIMEOUT_FREQ AF_MAX_BLK_LEN AF_VOLT_SUP AF_MAX_CUR_33V AF_MAX_CUR_30V AF_MAX_CUR_18V AF_DMA_SUP AF_SD0_EN AF_SD0_8BIT_EN AF_SD0_PULL_UP_EN AF_SPEC_VER [7:0] AF_VEND_VER [7:0]	These bits are used to configure the SD/SDIO/CE-ATA core in the ASSP.	Set to the appropriate values if the SD/SDIO/CE_ATA is to be utilized. Set the following anti-fuse signals: AF_SPEC_VER = 0x01h AF_VEND_VER = 0x00h AF_MAX_BLK_LEN = 00b AF_VOLT_SUP = 001b

6.2.1 Multi-Function Pins

Table 6-4 and **Table 6-5** shows I/O pins on the ArcticLink device that have more than one function. These pins are normally GPIO pins unless the ULPI and/ SDIO functions are enabled via the corresponding configuration control registers. As shown in **Table 6-5**,ULPI mode is an option for GPIO bank D. This mode is selected when AF_ULPI_EXT_EN = 1.

© 2007 QuickLogic Corporation

As shown in **Table 6-4**, SDIO mode is an option for GPIO bank C, this mode is active if AF_SD0_EN=1 and AF_SD0_8BIT_EN=1. If AF_SD0_EN=1 and AF_SD0_8BIT_EN=0, then SD0_DATA[4:7] can be used as regular GPIO pins.

Table 6-4: Multi-Function Pins for GPIO Bank C

PU121 Pin Number	PT196 Pin Number	I/O Bank	Direction	SDIO Mode	GPIO Mode
H2	L1	C	I/O	ASSP_SD0_DATA(0)	GPIO(C)
H1	K2	C	I/O	ASSP_SD0_DATA(1)	GPIO(C)
G2	K1	C	I/O	ASSP_SD0_DATA(2)	GPIO(C)
H3	J2	C	I/O	ASSP_SD0_DATA(3)	GPIO(C)
F3	H3	C	I/O	ASSP_SD0_DATA(4)	GPIO(C)
F1	G1	C	I/O	ASSP_SD0_DATA(5)	GPIO(C)
E2	G2	C	I/O	ASSP_SD0_DATA(6)	GPIO(C)
E3	F4	C	I/O	ASSP_SD0_DATA(7)	GPIO(C)
G1	H1	C	O	ASSP_SD0_CLK	GPIO(C)
G3	H2	C	I/O	ASSP_SD0_CMD	GPIO(C)

Table 6-5: Multi-Function Pins for GPIO Bank D

PU121 Pin Number	PT196 Pin Number	I/O Bank	Direction	ULPI Mode	GPIO Mode
B5	A1	D	I/O	ASSP_ULPI_DATA(0)	GPIO(D)
A5	B4	D	I/O	ASSP_ULPI_DATA(1)	GPIO(D)
C5	D5	D	I/O	ASSP_ULPI_DATA(2)	GPIO(D)
C4	C2	D	I/O	ASSP_ULPI_DATA(3)	GPIO(D)
B3	B3	D	I/O	ASSP_ULPI_DATA(4)	GPIO(D)
B4	D4	D	I/O	ASSP_ULPI_DATA(5)	GPIO(D)
C3	B2	D	I/O	ASSP_ULPI_DATA(6)	GPIO(D)
A3	C3	D	I/O	ASSP_ULPI_DATA(7)	GPIO(D)
A4	B1	D	I	ASSP_ULPI_CLK	GPIO(D)
D5	C4	D	I	ASSP_ULPI_DIR	GPIO(D)
C6	C5	D	I	ASSP_ULPI_NXT	GPIO(D)
A6	B5	D	O	ASSP_ULPI_STP	GPIO(D)

6.3 Designing with ArcticLink in QuickWorks

The following subsection explains how to implement ArcticLink using QuickLogic's libraries and software tools. When designing with ArcticLink, the ASSP core can be instantiated in schematic, Verilog, or VHDL.

6.3.1 File Types

There are three file types required for implementing ArcticLink designs:

- ArcticLink library files
- ArcticLink ASSP core
- Programmable logic design files

© 2007 QuickLogic Corporation

The library files define the various functional blocks available within an ArcticLink device which include RAM modules, FIFOs, CCMs, and QuickLogic macro blocks. All of the library files are located in the SpDE software's default installation directory `/pasic/spde/data`.

The following lists the library file types:

- Macros
- RAM Blocks
- FIFO Blocks
- Configurable Clock Managers (CCMs)

Both CCMs and ArcticLink specific macros are included in the macros pp.v and macros pp.vhd files.

RAM and FIFO blocks are generated in SpDE with the Tools > RAM ROM Wizard command. The resulting files are of the form r256x18 256x18.v where:

r = RAM (f = synchronous FIFO, af = asynchronous FIFO)

256 x 18 = write port depth 'x' width

512 x 9 = read port depth 'x' width

.v = Verilog (.vhd for VHDL)

The ArcticLink ASSP core can be instantiated with the ql1a100.v or ql1a100.vhd file. The ql1a100 file is only a wrapper file, so only a port map is defined. The port map is all that is required for synthesis since the ASSP core is instantiated in the SpDE software when the netlist is loaded.

6.3.2 Design using Schematics

QuickWorks is QuickLogic's development software suite which includes a schematic capture tool called *Schematic Editor*.

To open the Schematic Editor:

1. Open the development tool SpDE.
2. From the SpDE software, select the **Design->Schematic Editor & Navigator** command.
3. Click on "New" to start a new design. Executing this command will start the Hierarchy Navigator software, which is a tool used for viewing and traversing schematic designs.

To edit the design:

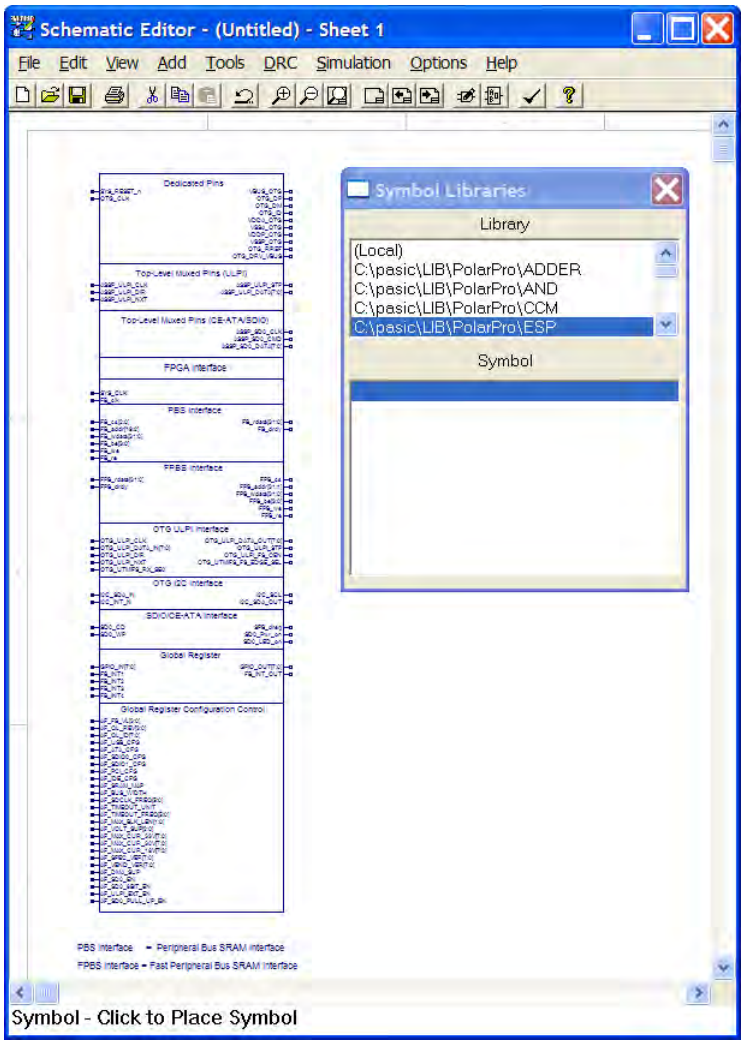
1. Select the **File->Create Schematic** command.
2. In order to instantiate the ArcticLink ASSP core, select **Add->Symbol** and browse down to the ESP library.
3. Left click on the desired package, i.e. QL1A100
4. To place the symbol on the sheet, drag the cursor to the schematic sheet and left click.

© 2007 QuickLogic Corporation

Once the ArcticLink symbol is placed, the corresponding connections can be made to other design symbols.

For further information on the Hierarchy Navigator and Schematic Editor, see the *QuickWorks User Manual* in the default installation directory `/pasic/spde/doc`.

Figure 6-1: ArcticLink Schematic Symbol



Once the schematic design is completed, it can be exported to Verilog or VHDL for synthesis by closing Schematic Editor, which returns to the Hierarchy Navigator, and issuing the **Tools->Export QuickLogic** command.

NOTE: If the schematic design is done entirely in schematic, it can be exported directly to an EDF netlist. An EDF file can then be directly loaded into the SpDE development software for placement and routing.

6.3.3 Design using Verilog/VHDL

The Verilog bus functional model (BFM)—QL1A100.v, can be used to instantiate the ArcticLink ASSP core.

The following is an excerpt of the Verilog instantiation of the ArcticLink ASSP core:

```
QL1A100 arcticlink_ASSP (
    .VBUS_ORG (vbus_org),
    .OTG_DP   (otg_dp),
    .OTG_DM   (otg_dm),
    .....
    .....
    .SD0_CD   (sd0_cd)
    );
```

The bus functional model (BFM) of the ArcticLink ASSP core is only available in Verilog. However, since Precision supports multi-language designs, the ASSP core can be instantiated in VHDL and then added to a Precision synthesis project as a Verilog file.

The following is an example of instantiating the Verilog ArcticLink ASSP core in VHDL:

```
arcticlink_isnt : QL1A100
    Port map (
        VBUS_ORG => vbus_org,
        OTG_DP   => otg_dp,
        OTG_DM   => otg_dm,
        .....
        .....
        SD0_CD   => sd0_cd
    );
```

6.3.4 Synthesizing ArcticLink Programmable Fabric Design

After an ArcticLink design is implemented in Verilog or VHDL code, it must be converted to a functional block netlist using the *Mentor Graphics Precision Synthesis* tool.

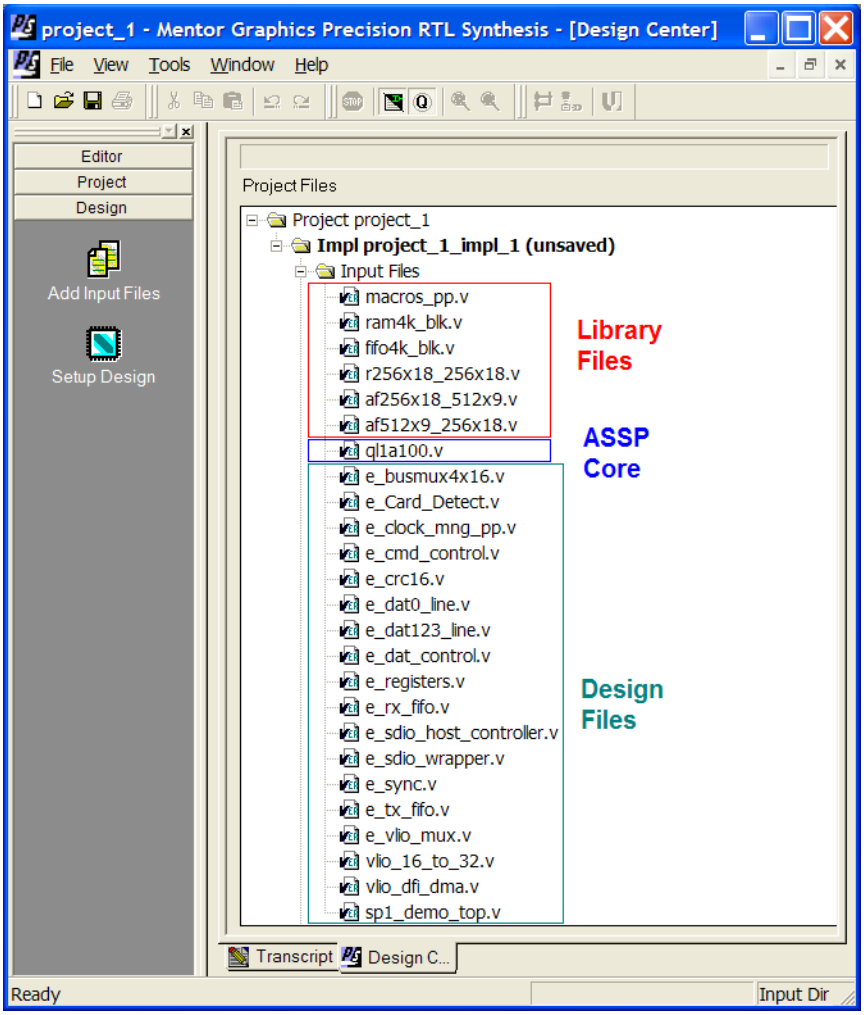
To open Precision Synthesis:

1. Open the development tool SpDE.
2. From the SpDE software, select the **File->Import Using Precision RTL** command.
3. From the Precision software, select the Design tab.
4. Click on **Add Input Files** to add all of the relevant files.
5. Select **Setup Design** and choose the desired ArcticLink device from the PolarPro family menu.
6. Run the Compile and Synthesize commands to generate the EDF netlist file.
7. Exit Precision and SpDE will automatically load the EDF file. If the EDF file is located in a different folder then you can select **File->Open** and browse to the location of the EDF file.

© 2007 QuickLogic Corporation

Precision automatically recognizes the compilation order of designs. However, it is always good engineering practice to specify the compilation order. **Figure 6-2** shows an example synthesis project with the files highlighted by file type.

Figure 6-2: ArcticLink Synthesis Example



Once the EDF file is loaded, SpDE will prompt the user to select the device and package. After the design loads, the chip design can be placed and routed by running the **Tools->Run Selected Tools** command.

6.4 Simulation Modeling

In order to aid engineers with their designs, specifically for those functional blocks that will be interfacing with the ASSP, there is a need to provide a simulation model for verification. For ArcticLink, a bus function model of the ASSP is required for HDL simulation and verification. The BFM is only available in Verilog so for designs written in VHDL, mixed-language synthesis and simulation tools are necessary. Fortunately, Precision Synthesis by

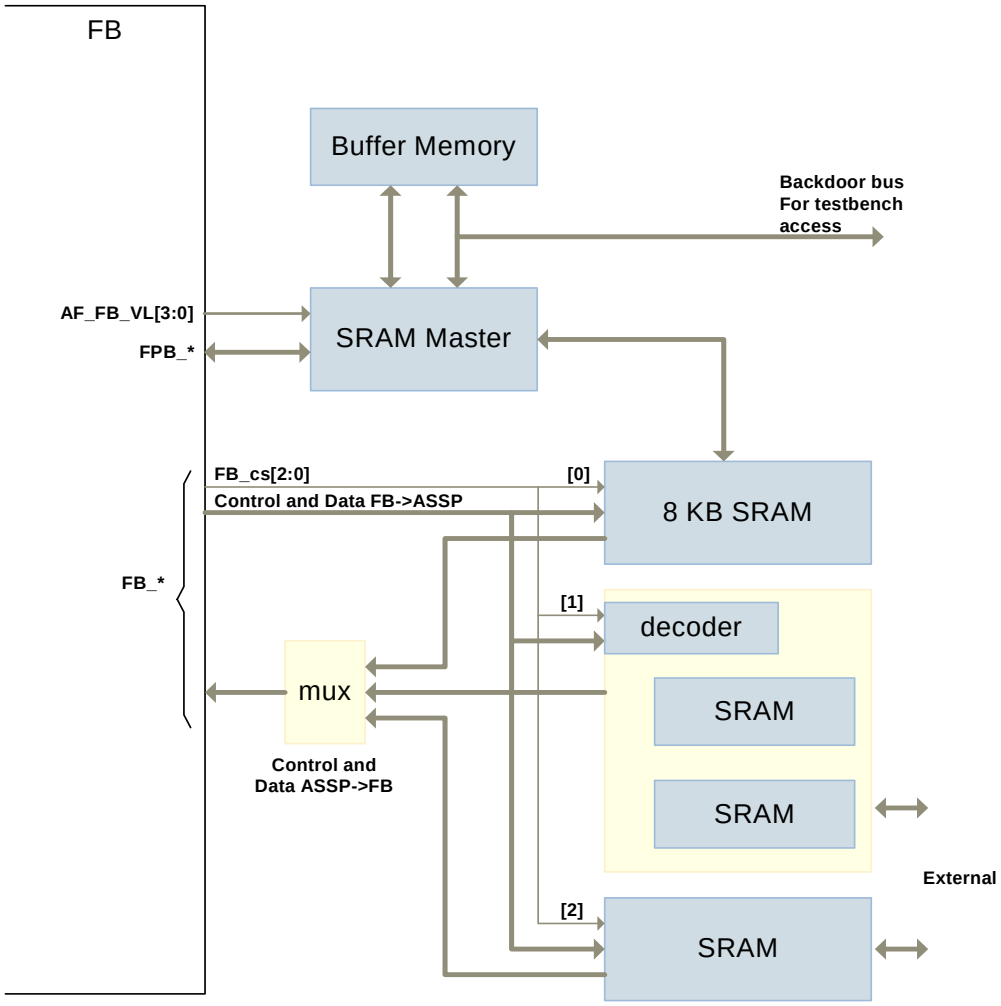
© 2007 QuickLogic Corporation

Mentor Graphics natively supports mixed language designs. The ArcticLink BFM can be acquired from one of QuickLogic's Regional Sales Managers. Contact information is available in the Contact Us section of the QuickLogic website at www.quicklogic.com.

6.4.1 ASSP/Fabric Interfaces

The ASSP/Fabric interfaces are comprised of two SRAM-like buses: a slave bus and a master bus. The slave bus allows an external master, designed either in the fabric or external to the chip, to access the different functional blocks within the ASSP through chip select regions and memory apertures. Likewise, the master interface provides a way for the internal DMA on the USB core to access external memory, either to extend the memory size or improve performance through bus mastering. The block diagram of the ASSP BFM, is provided, see **Figure 6-3**.

Figure 6-3: ASSP Bus Functional Model



© 2007 QuickLogic Corporation

In order to simplify the interface for the ArcticLink device, there is no special provision for the supporting bursts. Bursts can be achieved by doing back-to-back single accesses. The DRDY signal for read accesses is defined as an early ready signal in order to allow back-to-back reads from the 8 KB synchronous SRAM.

6.4.2 Slave Interface

The slave interface (ASSP to Fabric interface) allows designs external to the ASSP to access the different functional blocks in the ASSP. This external access will either be from a master designed into the programmable fabric or an external master, such as a processor, with a simple bridge to translate bus protocols. There are three chip selects provided to distinguish accesses to different blocks. One chip select is provided for the internal 8 KB SRAM block, one for the registers in the USB OTG core, and the last to the SDIO/CE-ATA and global registers.

Figure 6-4: Slave Interface Block Diagram

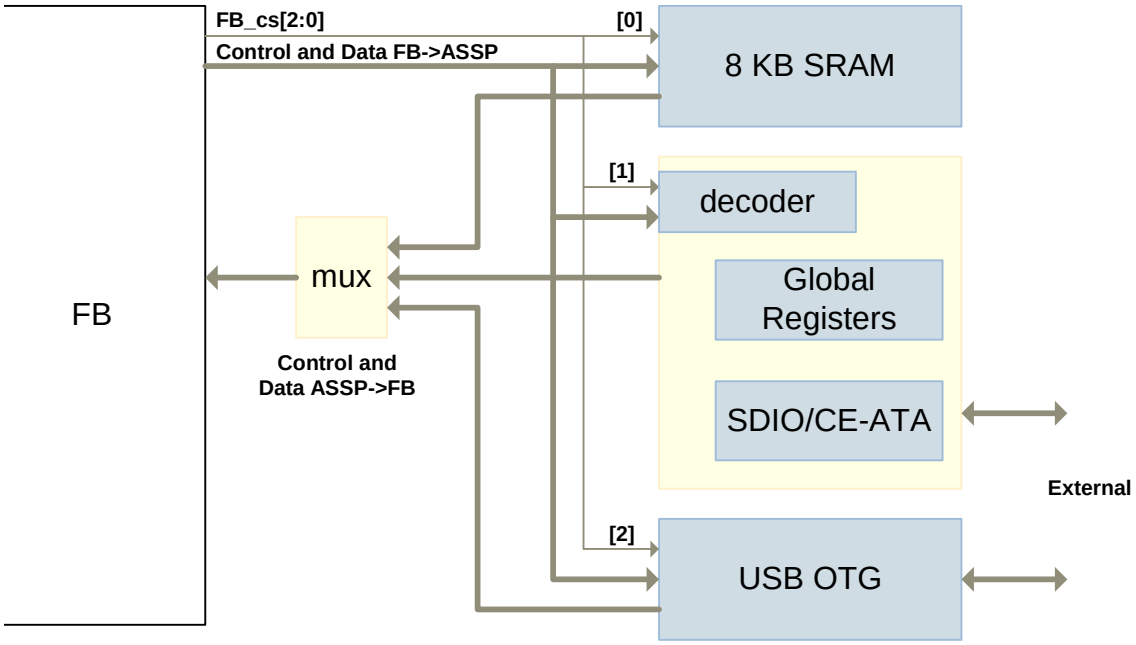


Table 6-6: Slave Interface Signal List

Pin Name	Dir	Width	Description	Default
Peripheral Bus => 1 signal (Common for Slave and Master)				
FB_clk	Out	1	SRAM Interface clock	1'b0
Peripheral Buses SRAM Interface => 91 signals				
FB_cs [2:0]	Out	3	Chip select (8 KB RAM, USB OTG Registers, CE-ATA/SDIO/Global Registers)	3'b0
FB_addr [19:2]	Out	18	Address (32-bit aligned access only, hsel to be decoded from this address)	18'b0
FB_wdata [31:0]	Out	32	Write data	18'b0
FB_rdata [31:0]	In	32	Read data	

© 2007 QuickLogic Corporation

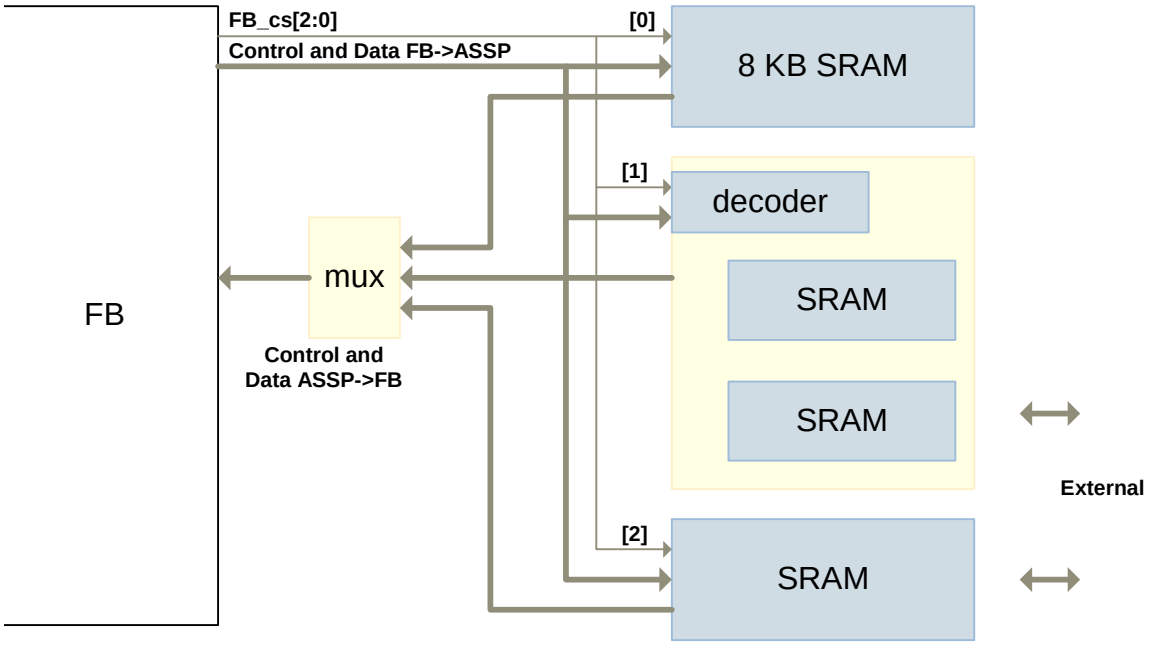
Table 6-6: Slave Interface Signal List

Pin Name	Dir	Width	Description	Default
FB_be [3:0]	Out	4	Byte enable	4'b0
FB_drdy	In	1	Data ready	
FB_we	Out	1	Write enable	1'b0
FB_re	Out	1	Read enable	1'b0

With the exception of the 8 KB SRAM block, accesses through the slave interface are register accesses. Note that this explanation is simply a model. In an actual design, a write to a register will set the value of the register, which ultimately causes the resulting event. In addition, reads return the value of a register, with an added possibility of triggering an event. For example, destructive reads on a data read register causes pops on the corresponding FIFO.

The BFM for the slave interface replaces all functional blocks with SRAM models. This provides a simple way of verifying the interface. However, the back-end portion of the interface is removed. This implies that the events that these registers cause, discussed in the previous section, will never happen. In addition, all SDIO, CE-ATA and USB transactions will never happen since signals are tied off inside the model. Below is a simple block diagram of the interface BFM, see **Figure 6-5**.

Figure 6-5: Slave Interface BFM



© 2007 QuickLogic Corporation

6.4.3 Master Interface

The master interface (Fabric to ASSP interface) of the ASSP is provided to allow DMA access to external memory. This access provides a way to increase memory space for the USB OTG core or permit bus mastering for systems that allow it for better performance.

Figure 6-6: Master Interface Block Diagram

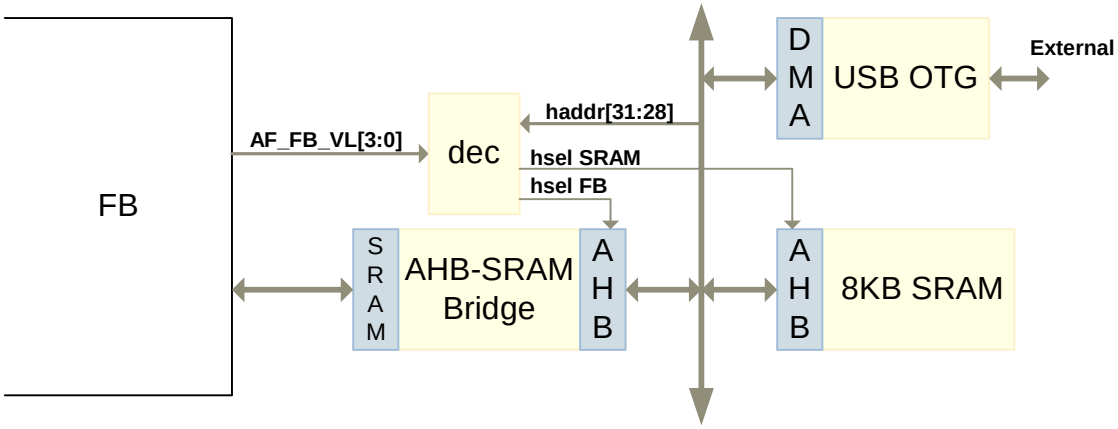


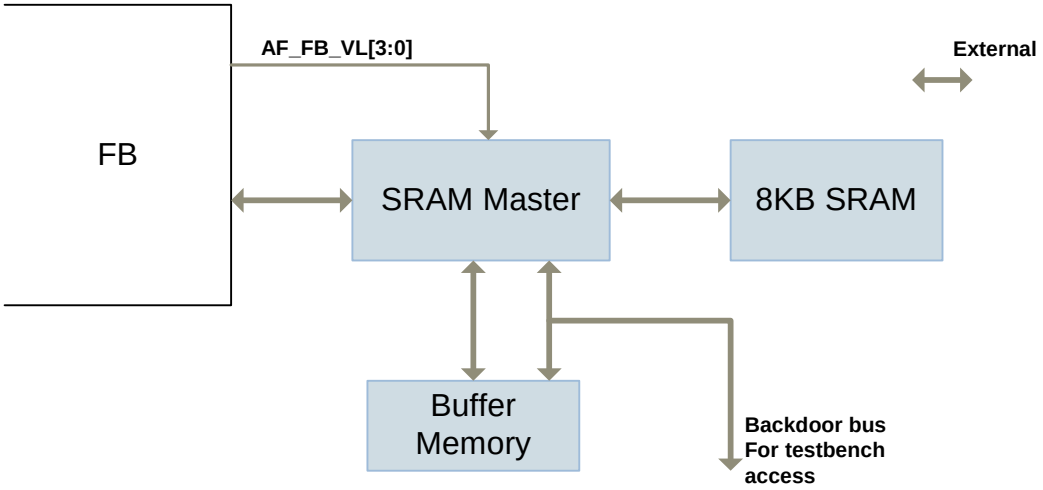
Table 6-7: Master Interface Signal List

Pin Name	Dir	Width	Description	Default
Peripheral Bus => 1 signal (Common for Slave and Master)				
FB_clk	Out	1	SRAM Interface clock	1'b0
Fast Peripheral Bus/ SRAM Interface (Master) => 104 signals				
FPB_cs	In	1	Chip select	
FPB_addr [31:1]	In	31	Address	
FPB_wdata [31:0]	In	32	Write data	
FPB_rdata [31:0]	Out	32	Read data	32'b0
FPB_be [3:0]	In	4	Byte enable	
FPB_we	In	1	Write enable	
FPB_re	In	1	Read enable	
FPB_drdy	Out	1	Data ready	1'b0

Based on the above block diagram, **Figure 6-6**, the BFM for this interface allows designers to program the master interface accesses the same way they program a DMA. It is important to note that it is also possible for simulations to be able to seed the DMA outgoing data and to be able to check incoming data for data corruption. This seeding and checking of data allows designers to verify their design thoroughly and effectively.

© 2007 QuickLogic Corporation

Figure 6-7: Master Interface BFM



6.4.4 Pre-layout Simulation

For pre-layout simulation—the bus functional model file (ql1a100.v) is compiled first, followed by the design files, followed by the testbench files.

The following files are included in the ArcticLink bus functional model package from QuickLogic:

- ql1a100.v – BFM file
- sb_master.v – Processor model. This file includes tasks to read, write, etc.
- tb.v – Sample testbench
- defines.v – Generated by SpDE during back annotation. The existing file will be overwritten if the design is re-run through the SpDE tools.
- *.do – Sample macro scripts for Aldec and Model Sim pre-layout and post-layout simulation.

The BFM has the same name as the ArcticLink synthesis wrapper file, so make sure to include the ArcticLink BFM and **not** the wrapper file. The BFM must be used otherwise no functional behavior will be displayed during simulation.

6.4.5 Post-layout Simulation

For post-layout simulation, the order of compilation is similar to pre-layout, except the design files will be replaced with the post-layout netlist files (*.vq/*.vhq, where "*" is the name of the top level HDL file). SpDE will generate a custom defines.v file based upon the speed grade, operating range, and temperature corner selected by the user in the **Tools>Options>Delay Modeler** menu. The file will contain a series of define statements which contain the timing information (Kv and Kvt) and will be included in the BFM file during compilation. Therefore, it is important to compile the BFM file (ql1a100.v) with the **DEFINE** option in order for the timing information to be passed from SpDE to the BFM.

© 2007 QuickLogic Corporation

Chapter 7
Design Considerations

This chapter explains the various ArcticLink interface blocks, and how to integrate the embedded ArcticLink features into specific designs. Example configurations and applications are given.

This chapter contains the following sections:

- **"ArcticLink Design Interfaces" on page 57**
- **"Application Models" on page 59**

7.1 ArcticLink Design Interfaces

The following subsections explain the various interfaces available within the ArcticLink architecture.

7.1.1 ASSP/Programmable Fabric Interface Ports

Depending upon what features are utilized in the ArcticLink ASSP core, the corresponding antifuse static configuration signals need to be assigned. There are configuration signals for the USB OTG block, the SDIO block, and the overall ASSP core system. See QuickLogic's *ArcticLink Solution Platform Data Sheet* for detailed descriptions on the fabric interface ports.

7.1.2 Clocking Schemes and CCM Usage

The ArcticLink embedded global clock network and Configurable Clock Manager (CCM) combination is ideal for speed optimization and implementing multiple clock domain designs. Each quadrant of the programmable fabric includes five global clock buffers for reset signals, clocks, and fast-propagation low-skew critical signal paths. For designs that require a strong stable clock signal or some multiple of an existing input clock, the CCM can be utilized. The CCM has three modes of operation, based on the input frequency and desired output frequency as shown in **Table 7-1**.

Table 7-1: CCM Mode Frequencies

Output Frequency	Input Frequency Range	Output Frequency Range	PLL Mode
x1	25 MHz to 200 MHz	25 MHz to 200 MHz	PLL_MULT1
x2	25 MHz to 100 MHz	50 MHz to 200 MHz	PLL_MULT2
x4	25 MHz to 50 MHz	100 MHz to 200 MHz	PLL_MULT4

In addition, pllout0 has a 0° phase shift and pllout1 has an optional 0°, 90°, 180°, or 270° phase shift plus a programmable delay up to 2.5 ns at 250 ps intervals.

© 2007 QuickLogic Corporation

See QuickLogic's *Applications Note 92: Clock Networks in the ArcticLink Solutions Platform* for more information on the available clock resources and Configurable Clock Manager.

7.1.3 Peripheral vs. Fast Peripheral Interface

There are two peripheral buses connecting the ArcticLink ASSP core to and from the programmable fabric. The two buses are the Peripheral Bus and the Fast Peripheral Bus. Essentially, the Peripheral bus interfaces between the programmable fabric to the ASSP core, where the programmable fabric is the master for managing traffic between the two functional blocks. Conversely, the Fast Peripheral Bus specifies the ASSP core as the master.

The Fast Peripheral bus is useful for improving performance by providing a way for the internal DMA engine in the Hi-Speed USB 2.0 OTG controller to directly access external devices such as memory through an interface controller implemented in the fabric. This type of implementation is advantageous since this requires no direct control from an external controller like an applications processor. For further information about the Peripheral Bus and the Fast Peripheral Bus see the *ArcticLink Solution Platform User Manual*.

7.1.4 Top-level Multi-function Pins

There are specific pins on the ArcticLink device that have more than one function. These pins are normally general purpose I/O pins. However, depending upon what features are used in the ArcticLink ASSP core, these pins can be configured to connect from an ASSP functional block to an I/O pad. These pins are configured via the Global Register Configuration Control registers.

See QuickLogic's *ArcticLink Solution Platform Data Sheet* for detailed descriptions and locations of the multi-function pins.

© 2007 QuickLogic Corporation

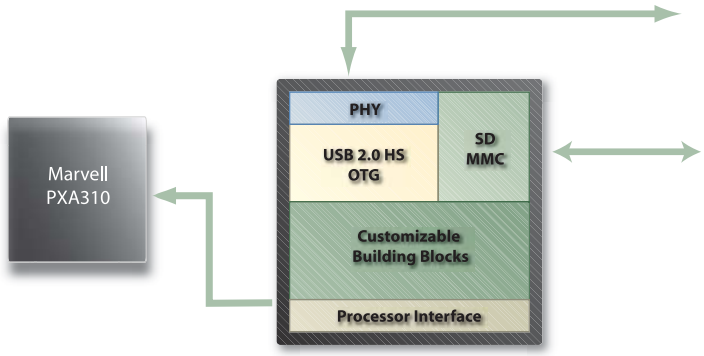
7.2 Application Models

7.2.1 Marvell Monahan DFI Interface

QuickLogic's ArcticLink has embedded programmable fabric, which allows designers to interface to a broad variety of host controllers around storage, networking, video and other high-speed peripherals including but not limited to USB 2.0 OTG with PHY, SD/SDIO/MMC, IDE, CE-ATA, NAND/Managed NAND, Hi-Speed UARTs, Cardbus, and PCI. Additionally, these functional blocks can be interfaced t a variety of applications processors. One example is QuickLogic's applications processor IP to Marvell's Monahan family of Processors via the DFI interface bus.

This particular bridging connectivity solution has the advantage of low power, performance, small form factor, and low cost. Furthermore, since the solution was optimized for minimal resource utilization, a substantial amount of the programmable fabric is available of further IP development and interfaces.

Figure 7-1: Marvell PXA320 Monahan Processor Interface



7.2.2 Memory DMA and Bus Mastering

ArcticLink can be utilized to not only bridge an applications processor to some fixed functional block, but also to master direct access to functional blocks without interrupting the processor. As an exmaple, **Figure 7-2** shows an application where a processor is connected to a WiFi chipset via PCI with addtional support for SDRAM access. In this case, ArcticLink has direct access to the SDRAM and processor. The advantage of this configuration is that the memory bus can be mastered by the ArcticLink device to initiate DMA access directly to the SDRAM without the requirement of the processor. This architecture offers the added benefit of higher bandwidth and data transfers without active involvement of the processor.

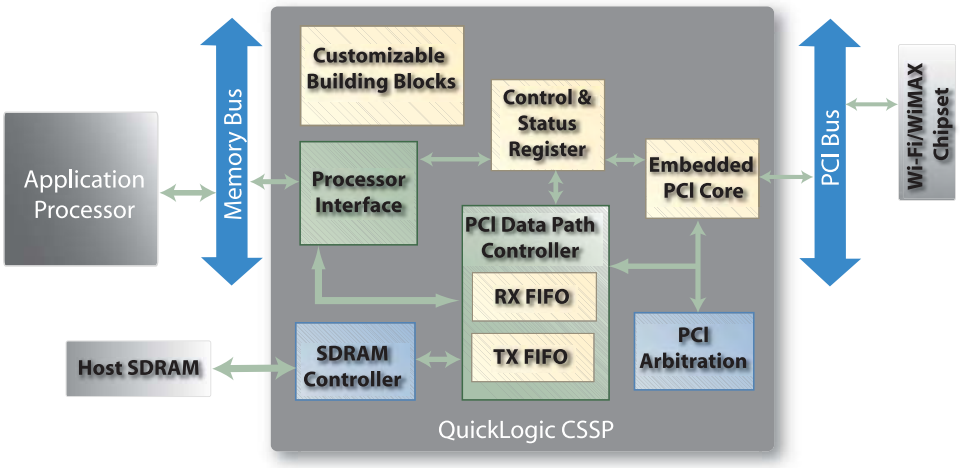
The SDRAM interface arbitrates for the host processor memory bus, and controls the host SDRAM during transfers initiated from PCI. When a device on PCI tries to write to host memory, the SDRAM controller module is activated to pop data from the Receive FIFO of the PCI data path controller into host SDRAM memory. When a device on PCI tries to read from host memory, the SDRAM controller module is activated to push data into the

© 2007 QuickLogic Corporation

Transmit FIFO of the PCI data path controller from host SDRAM memory. IF desired, the processor can even control the burst behavior of the SDRAM interface via ArcticLink configuration registers.

PCI initiated SDRAM accesses have the ability to DMA data to and from the host memory. The ArcticLink bridge will arbitrate for control of the processor memory bus, which is transparent to the PCI devices.

Figure 7-2: SDRAM Bus Mastering



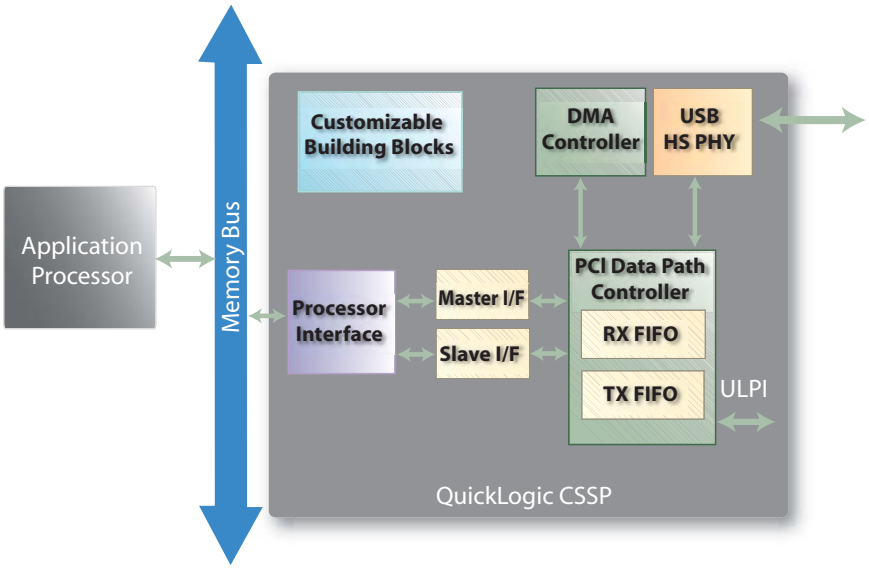
7.2.3 USB Enumeration

Whenever a USB device is connected to a USB host, the device goes through a mandatory enumeration sequence. In embedded applications, the microprocessor is generally included in the handling of the enumeration process. Unfortunately, if the processor is in a low power standby mode, then this requires the wake-up sequence of the processor, which consumes time, processor utilization, and power. The ArcticLink USB OTG solution allows for the ArcticLink to handle the entire enumeration process without having to interrupt the processor.

Upon insertion of a new device, the USB host establishes a communications path between the host and the device. The host then enumerates the device by sending control transfers consisting of standard USB requests. Subsequently, the device responds to each request by returning the requested information and taking the corresponding actions.

© 2007 QuickLogic Corporation

Figure 7-3: USB OTG 2.0 Solution



7.2.4 USB Hub

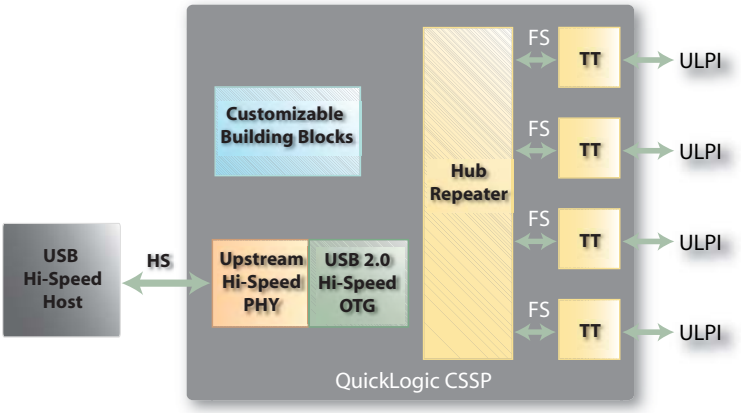
The USB specification defines a hub topology to support port expansion. A USB hub has one upstream root port and multiple downstream peripheral ports. All peripherals connected to the downstream ports share the same bandwidth provided by the upstream port. QuickLogic's ArcticLink solution platform provides a single Hi-Speed USB OTG port. If a given system requires multiple USB connections, the low power customizable building blocks can be used to implement a multi-port USB hub. The exact number of downstream ports can be configured based on customer specific requirements to save resources and power. Each downstream port can support Full-Speed USB peripherals.

The USB Hub solution supports:

- Hi-Speed USB upstream port with built-in Hi-Speed USB PHY
- Each downstream port supports Full-Speed (12 Mbits/s) and Low-Speed (1.5 Mbits/s) operation
- A customizable number of downstream ports
- Transaction Translator (TT) optimized for maximized bandwidth utilization

© 2007 QuickLogic Corporation

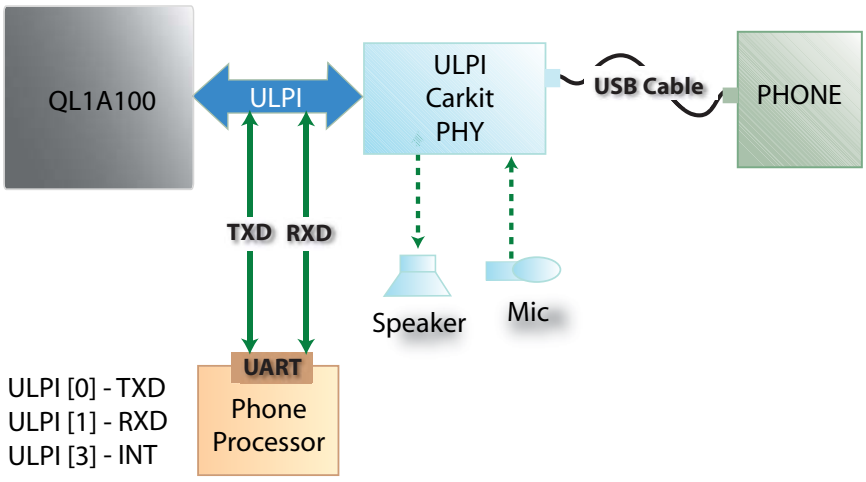
Figure 7-4: USB Hub



7.2.5 Carkit Support in Programmable Fabric

Typically, USB-ULPI Carkit PHYs support three main modes HS USB, UART and Audio. Implementing a carkit application in the programmable fabric of an ArcticLink device can support the USB responsibility of multiplexing the ULPI data lines between UART and USB-ULPI modes, and receiving carkit interrupts from the PHY. In conjunction with the ArcticLink USB 2.0 OTG ASSP core, the programmable fabric can support a full-speed CEA-936-AI^{2}C carkit interface.**Figure 7-5** shows the ArcticLink Solution Platform in an in-dash carkit. For more information see *Carkit Support* in the *ArcticLink Solution Platform Data Sheet*.

Figure 7-5: In-Dash Carkit Application Block Diagram



© 2007 QuickLogic Corporation

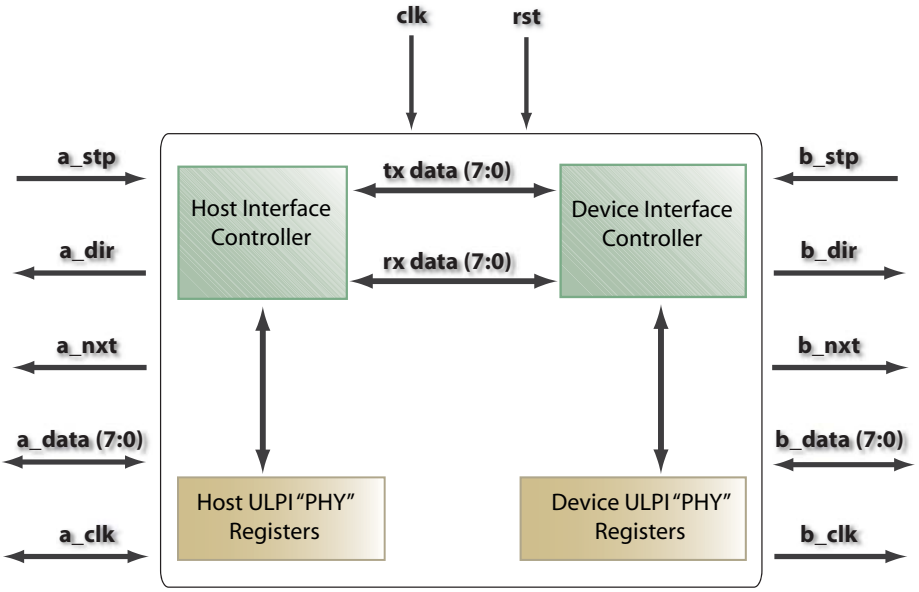
7.2.6 ULPI PHY Impersonator in Programmable Fabric

Most recently, USB has been widely adopted by mobile devices as the high-speed interface used for multimedia, data transfer and synchronization. The latest USB 2.0 standard supports Hi-Speed with up to 480 Mbits/s throughput. The On-the-Go (OTG) capability allows two devices to connect to each other without the use of a computer.

As USB becomes more ubiquitous, system designers have also adopted it for chip-to-chip interconnect. Traditionally, for two chips to communicate with each other, each would require a PHY—one for the Host port and the other for the Device port. A PHY contains digital processing logic and an analog transceiver. Running at high speed, a PHY consumes power at well over 50 mA. The requirement to have PHYs on both sides of the connections leads to higher cost, space and most importantly power consumption— which is a big concern for these power sensitive mobile services.

QuickLogic's ArcticLink provides the unique capability of implementing a pair of PHY impersonators in the programmable fabric. This PHY-less ULPI block acts as a bridge to communicate between a ULPI Host link and a ULPI Device link at high speed without the high power transceivers. The PHY-less ULPI emulates actual PHY functionality—which includes: ULPI PHY register read and write accesses, RXCMD, and TXCMD—without actually having to go through D+/D- on the USB wires, making this block ideal for chip-to-chip interconnect in a mobile device.

Figure 7-6: PHY-less ULPI Bridge



© 2007 QuickLogic Corporation

Chapter 8
Control and Status Registers

This chapter contains the following sections:

- ArcticLink System
  - **"Memory Map" on page 66**
  - **"Common Registers" on page 67**
- SD/SDIO/MMC and CE–ATA Register Descriptions
  - **"SD/SDIO and CE–ATA Registers" on page 73**
  - **"CE-ATA Control Register (Offset 100h)" on page 77**
- USB OTG Control and Status Registers
  - **"USB OTG Control and Status Overview" on page 77**
  - **"CSR Memory Map" on page 78**
  - **"Register Descriptions" on page 86**

© 2007 QuickLogic Corporation

8.1 ArcticLink System

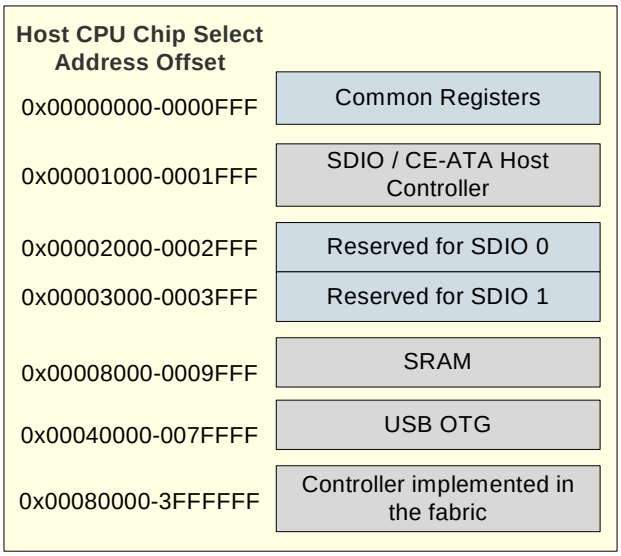
8.1.1 Memory Map

The ArcticLink Solution Platform is connected to a Chip Select of the external host processor (i.e. Marvell PXA). The ArcticLink Solution Platform has a fixed memory map for the peripherals embedded in the ASSP. The full memory map is constructed when the host CPU interface is implemented in the ArcticLink's Programmable Fabric.

A recommended memory map of the ArcticLink Solution Platform is shown, see **Figure 8-1**. All addresses contained in the memory map are offsets from the base address of the Chip Select used. Configure the chip selects, FB_cs[2:0], before attempting any access to the ArcticLink device.

NOTE: For the Marvell PXA27x 26 bits of address space are available.

Figure 8-1: Chip Select Memory Map for the ArcticLink Device



8.1.1.1 Register Application Access Types

All registers are 32 bits in data width. The register read and write accesses will always be 32 bits in width. All byte enables must be enabled for register accesses. The byte enables may, however, be used to control the data bus widths to the on-chip SRAM memory.

The Access column of each register description that follows specifies how the application, and the Hi-Speed USB 2.0 OTG core can access the register fields. **Table 8-1** lists abbreviations for all register types.

Table 8-1: Abbreviations Used for Registers

Type	Description
Read Only (RO)	Register field can only be read by the application. Writes to read-only fields have no effect.
Write Only (WO)	Register field can only be written by the application.
Read and Write (RW))	Register field can be read and written by the application. The application can set or clear the bits to the desired state.

© 2007 QuickLogic Corporation

Table 8-1: Abbreviations Used for Registers (Continued)

Read, Write, and Self Clear (R_W_SC)	Register field can be read and written by the application (Read and Write), and is cleared to 1'b0 by the core (Self Clear). The conditions under which the core clears this field are explained in detail in the field's description.
Read, Write, Self Set, and Self Clear (R_W_SS_SC)	Register field can be read and written by the application (Read and Write), set to 1'b1 by the core on certain USB events (Self Set), and cleared to 1'b0 by the core (Self Clear). The conditions under which the core sets and clears this field are explained in the field's description. (Only the Port Resume bit of the Host Port Control and Status register, HPRT.PrtRes, uses this access type).
Read, Self Set, and Write Clear (R_SS_WC)	Register field can be read by the application (Read), can be set to 1'b1 by the core on a certain internal or USB or Internal Bus event (Self Set), and can be cleared to 1'b0 by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 has no effect on this field. The conditions under which the core sets this field are explained in detail in the field's description. (For example, interrupt bits.)
Read, Write Set, and Self Clear (R_WS_SC)	Register field can be read by the application (Read), can be set to 1'b1 by the application with a register write of 1'b1 (Write Set), and is cleared to 1'b0 by the core (Self Clear). The application cannot clear this type of field, and a register write of 1'b0 to this bit has no effect on this field. The conditions under which the core clears this field are explained in detail in the field's description. (For example, reset signals)
Read, Self Set, and Self Clear or Write Clear (R_SS_SC_WC)	Register field can be read by the application (Read), can be set to 1'b1 by the core on certain internal or USB or Internal Bus events (Self Set), and can be cleared to 1'b0 either by the core itself (Self Clear) or by the application with a register write of 1'b1 (Write Clear). A register write of 1'b0 to this bit has to no effect on this field. The conditions under which the core sets or clears this field are explained in the field's description. (Only the Port Enable bit of the Host Port Control and Status register, HPRT.PrtEna, and the VStatus Done bit of the PHY Vendor Control register, GPVNDCTL.VStsDone, use this access type.)

NOTE: Program Reserved fields with 0s. Read values from Reserved fields are unknowns (Xs).

8.1.2 Common Registers

The individual registers which comprise the set of Common registers in the ArcticLink Solution Platform are summarized in **Table 8-2**

NOTE: The Common Register block will have an Identification (ID) register at offset zero in the ArcticLink's recommended memory map,**Figure 8-1**.

Table 8-2: Common Register Set

Address [Host CPU Chip Select]	Register Name			
	Bit31 ------ Bit24	Bit23 ------ Bit16	Bit15 ------ Bit8	Bit7 ------ Bit0
000000h	Board ID			
000004h	Feature			
000008h	Interrupt Enable			
00000Ch	Interrupt Status			
000010h	Enable			
000014h	USB OTG Mapping			
000018h	System Clock Enable			
00001Ch – 000FFFh	Reserved			

© 2007 QuickLogic Corporation

8.1.2.1 ID Register (Offset Address: 000000h)

This register is used by software to identify the IP (see **Table 8-3**) of the ArcticLink Solution Platform plus fabric design. Each IP has a Revision number and a unique Device ID value that will be hard coded into this register (through vialink). This register also contains 8 GPIO bits.

NOTE: The GPIOs are optional

Table 8-3: ID Register

Name	Bit(s)	Type	Reset Value	Function
–	31:20	–	–	Reserved
GPIO	19:12	R/W	–	Read Value = TBD Write Value =TBD General Purpose Input/Output pins
Revision	11:8	RO	TBD	All devices created using the ArcticLink will have a Revision used by software to identify the version of the device. AF_QL_REV[3:0] = QuickLogic Revision code
Device ID	7:0	RO	TBD	AF_QL_ID[7:0] = QuickLogic ID code

8.1.2.2 Feature Register (Offset Address: 000004h)

This register is used by software to identify IP features (see **Table 8-4**) present in the specific design.

Table 8-4: Feature Register

Name	Bit(s)	Type	Reset Value	Function
–	31:9	–	–	Reserved
USB_OTG_PRSNT	8	RO	1	0 = no USB OTG 1 = USB OTG Present Indicates that the device has a USB OTG port. AF_USB_CFG=USB OTG configuration bit
CE_ATA_PRSNT	7	RO	1	0 = no CE-ATA 1 = CE-ATA Present Indicates that the device has a CE-ATA port. AF_ATA_CFG=CE-ATA configuration bit
SDIO_1_PRSNT	6	RO	0	0 = No SDIO 1 1 = SDIO 1 Present Indicates that the device has support for a Host SDIO connection AF_SDIO1_CFG=SDIO 1 configuration bit
SDIO_0_PRSNT	5	RO	0	0 = No SDIO 0 1 = SDIO 0 Present Indicates that the device has support for a Host SDIO connection AF_SDIO0_CFG=SDIO 0 configuration bit

© 2007 QuickLogic Corporation

Table 8-4: Feature Register (Continued)

Name	Bit(s)	Type	Reset Value	Function
–	4:2	–	–	Reserved
SRAM_PRSNT	1	RO	1	0 = no SRAM 1 = SRAM Present Indicates that the device directly maps the SRAM into the CPUs memory space AF_SRAM_MAP= SRAM address map bit
BUS_WIDTH	0	RO	–	0 = Host CPU is in 16-bit mode 1 = Host CPU is in 32-bit mode Bus Width AF_BUS_WIDTH= Host CPU bus width

8.1.2.3 Interrupt Enable Register (Offset Address: 000008h)

This read-write register is used to control which functions can output their interrupt to the interrupt pin (see **Table 8-5**).

Table 8-5: Interrupt Enable Register

Name	Bit(s)	Type	Reset Value	Function
GENERIC_INT_EN[3:0]	31:28	RW	0	0 = Interrupt Disabled 1 = Interrupt Enabled Interrupt enable bits for interrupt sources from the fabric
–	27:9	–	–	Reserved
USB_OTG_INT_EN	8	RW	0	0 = Interrupt Disabled 1 = Interrupt Enabled USB OTG Controller interrupt enable
CE_ATA_INT_EN	7	RW	0	0 = Interrupt Disabled 1 = Interrupt Enabled CE-ATA controller interrupt enable
SDIO_1_INT_EN	6	–	–	Not Available
SDIO_0_INT_EN	5	–	–	Not Available
–	4:0	–	–	Reserved

© 2007 QuickLogic Corporation

8.1.2.4 Interrupt Status Register (Offset Address: 000000Ch)

This read-only register is used to indicate which function is generating an interrupt. If a bit is high in this register and the corresponding bit in the Interrupt Enable register is high then the interrupt pin of the chip is driven (see **Table 8-6**).

Table 8-6: Interrupt Status Register

Name	Bit(s)	Type	Reset Value	Function
GENERIC_INT_STAT[3:0]	31:28	R	0	0 = No Interrupt 1 = Interrupt Active Interrupt status bits for interrupt sources from the fabric
–	27:9	–	–	Reserved
USB_OTG_INT_STAT	8	R	0	0 = No Interrupt 1 = Interrupt Active USB OTG Controller interrupt status
CE_ATA_INT_STAT	7	R	0	0 = No Interrupt 1 = Interrupt Active CE-ATA controller interrupt status
SDIO_1_INT_STAT	6	–	–	Not Available
SDIO_0_INT_STAT	5	–	–	Not Available
–	4:0	–	–	Reserved

8.1.2.5 Enable Register (Offset Address: 0000010h)

This read-write register provides a means for software to enable individual controllers contained within the device (see **Table 8-7**).

Table 8-7: Enable Register

Name	Bit(s)	Type	Reset Value	Function
–	31:19	–	–	Reserved
GL_OC_PHY_SUSPEND	18	R/W	0	0 = Not suspend 1 = Suspend On-chip global USB OTG PHY Suspend
VBUS_COMP_DISABLE	17	R/W	0	0 = Enabled 1 = Disabled VBUS Comparator Disable for the on-chip USB OTG PHY
ONBIST	16	R/W	0	0 = Disable BIST 1 = Enable BIST ON BIST Enable for USB OTG PHY
–	15:5	–	–	Reserved
USB_OC_PHY_EN	4	R/W	0	0 = USB OTG PHY Disabled 1 = USB OTG PHY Enabled USB On-Chip UTMI PHY Enable
USB_EN	3	R/W	0	0 = USB OTG Disabled 1 = USB OTG Enabled USB OTG Controller Enable

© 2007 QuickLogic Corporation

Table 8-7: Enable Register (Continued)

Name	Bit(s)	Type	Reset Value	Function
CE_ATA_EN	2	R/W	0	0 = CE-ATA Disabled 1 = CE-ATA Enabled CE-ATA or SD/SDIO Enable
SDIO_1_EN	–	–	–	Reserved
SDIO_0_EN	–	–	–	Reserved

The following sequence of operations assumes that the PHY is to be suspended with no wakeup capabilities (PHY in low power mode). The PHY must be disabled when the user switches to/from the ULPI interface or other USB interfaces.

1. Normal Mode with active UTMI interface
   - Set the StopPclk bit in the PCGCCTL register to suspend the UTMI PHY.
   - Set the VBUS_COMP_DISABLE high to disable the VBUS comparator in the on-chip UTMI PHY.
2. Normal Mode with inactive UTMI interface
   - Set the StopPclk bit in the PCGCCTL register to suspend the UTMI PHY.
   - Set the GL_OC_PHY_SUSPEND high to shunt the on-chip UTMI PHY suspend from the core.
   - Set the VBUS_COMP_DISABLE high to disable the VBUS comparator in the on-chip UTMI PHY.

8.1.2.6 USB OTG Mapping Register (Offset Address: 0000014h)

This read-only register provides control over the address the SRAM and Fabric Slave will be mapped to in the USB OTG controller's DMA engine (see **Table 8-8**).

Table 8-8: Enable Register

Name	Bit(s)	Type	Reset Value	Function
MAP_ADDR	31:28	R	see footnote[a]	Internal Bus 0 Map Address The Internal Bus 0 address decoder will compare the MSB of the Internal Bus 0 addresses [31:28] with these bits to enumerate the HSEL for the 16 KB (or 8 KB) SRAM and the Fabric Slave.
–	27:0	–	–	Reserved

a. These bits are hard-coded in the Fabric using vialink

8.1.2.7 System Clock Enable Register (Offset Address: 0000018h)

This read-write register provides a method for enabling and disabling clocks to the individual controllers contained within the device (see **Table 8-9**).

Table 8-9: Enable Register [a]

Name	Bit(s)	Type	Reset Value	Function
–	31:5	–	–	Reserved
OC_PHY_CK_EN	4	R/W	0	0 = Disable clock 1 = Enable clock USB On-Chip UTMI PHY Clock Enable
SYS_CK_EN	3	R/W	0	0 = Disable clock 1 = Enable clock System Clock Enable (Internal Bus Clock)
CE_ATA_CK_EN	2	R/W	0	0 = Disable clock 1 = Enable clock CE-ATA Clock Enable
SDIO_1_CK_EN	1	–	–	Not Available
SDIO_0_CK_EN	0	–	–	Not Available

a. This register is used to gate the clock in the previously mentioned IP blocks. The USB OTG core already has a register that can enable/disable the clock(s).

© 2007 QuickLogic Corporation

8.2 SD/SDIO/MMC and CE–ATA Register Descriptions

8.2.1 SD/SDIO/MMC/CE-ATA Host Controller Register Set

QuickLogic's SD/SDIO/MMC/CE-ATA Host Controller is fully compatible with the *SD Host Controller Specification v2.0*. **Table 8-10** summarizes the SD/SDIO/MMC/CE-ATA Host Controller register set.

NOTE: The SD/SDIO/MMC/CE-ATA Host Controller Register block will have an offset 01000h in the ArcticLink's recommended memory map, **Figure 8-1**.

Table 8-10: SD/SDIO/MMC/CE-ATA Host Controller Register Set

Address [25:0]	Register Name			
	Bit31 ------ Bit24	Bit23 ------ Bit16	Bit15 ------ Bit8	Bit7 ------ Bit0
000000h	Reserved			
000004h	Block Count		Block Size	
000008h	Argument			
00000Ch	Command		Transfer Mode	
000010h	Response bit 31 - 0			
000014h	Response bit 63 - 32			
000018h	Response bit 95 - 64			
00001Ch	Response bit 127 - 96			
000020h	Data Port			
000024h	Present State			
000028h	Wakeup Control	Block Gap Control	Power Control	Host Control
00002Ch	Software Reset	Timeout Control	Clock Control	
000030h	Error Interrupt Status		Normal Interrupt Status	
000034h	Error Interrupt Status Enable		Normal Interrupt Status Enable	
000038h	Error Interrupt Signal Enable		Normal Interrupt Signal Enable	
00003Ch	Reserved		Auto CMD12 Error Status	
000040h	Capabilities			
000044h	Capabilities (Reserved)			
000048h	Maximum Current Capabilities			
00004Ch	Maximum Current Capabilities (Reserved)			
000050h	Force Event for Error Interrupt Status		Force Event for Auto CMD12 Error Status	
000054h – 0000F8h	Not Defined			
0000FCh	Host Controller Version		Slot Interrupt Status	
000100h	CE-ATA Control			
000104h – 0001FFFh	Not Defined			

NOTE: For more details on register descriptions and operation sequences, refer to *SD Specifications Part A2: SD Host Controller Standard Specification*.

8.2.2 SD/SDIO and CE–ATA Registers

For specific information about the SD/SDIO registers refer to the *SD Host Controller Specification – SD Host Standard Register.*

The CE-ATA registers will be the same as the standard SD Host Controller register set with some extra bits given here, and an additional register for CE-ATA specific support, see **Table 8-17**.

© 2007 QuickLogic Corporation

SDA provides the standard SD Host Controller Specification. If you are looking for a copy of the specification visit the SD Card Association website at www.sdcard.org/HostController/index.html to download a pdf of the SD Host Controller Standard Simplified Specification Version 2.0.

8.2.3 Host Control Register (Offset 28h)

This register is defined in the SD Host Controller Specification. The bit 5 definition for an 8-bit Data Bus Width control function is provided.

Table 8-11: Host Control Register (Offset 28h)

Bit	Signal Name	Type	Reset Value	Function
7:6	Defined in SD Host Controller Specification			
5	DAT_8BIT_EN	R/W	0	8 bit Data Transfer Width 1= Enabled 0 = Disabled
4:2	Defined in SD Host Controller Specification			
1	DAT_WIDTH	R/W	0	Data Transfer Width 1= 4-bit 0 = 1 or 8-bit (see bit 5)
0	Defined in SD Host Controller Specification			

8.2.4 Normal Interrupt Status Register (Offset 30h)

This register is defined in the SD Host Controller Specification. The bit 9 definition for the CE_ATA Command Completion Interrupt function is provided.

Table 8-12: Normal Interrupt Status Register (Offset 30h)

Bit	Signal Name	Type	Reset Value	Function
15:10	Defined in SD Host Controller Specification			
9	CEATA_CMD_CMPLT	R/W	0	CE-ATA Command Completion Interrupt Status 1= CE-ATA transfer completed 0 = CE-ATA transfer incomplete
8:0	Defined in SD Host Controller Specification			

© 2007 QuickLogic Corporation

8.2.5 Normal Interrupt Status Enable Register (Offset 34h)

This register is defined in the SD Host Controller Specification. The bit 9 definition for the CE_ATA Command Completion Interrupt Status Enable function is provided.

Table 8-13: Normal Interrupt Status Enable Register (Offset 34h)

Bit	Signal Name	Type	Reset Value	Function
15:10	Defined in SD Host Controller Specification			
9	CEATA_CC__STAT_EN	R/W	0	CE-ATA Command Completion Interrupt Status 1= ENABLED 0 = MASKED
8:0	Defined in SD Host Controller Specification			

8.2.6 Normal Interrupt Signal Enable Register (Offset 38h)

This register is defined in the SD Host Controller Specification. The bit 9 definition for the CE_ATA Command Completion Interrupt Signal Enable function is provided.

Table 8-14: Normal Interrupt Signal Enable Register (Offset 38h)

Bit	Signal Name	Type	Reset Value	Function
15:10	Defined in SD Host Controller Specification			
9	CEATA_CC__SIGN_EN	R/W	0	CE-ATA Command Completion Interrupt Signal 1= ENABLED 0 = MASKED
8:0	Defined in SD Host Controller Specification			

8.2.7 Capability Register (Offset 40h)

This register is defined in the SD Host Controller Specification. **Table 8-15** provides the the ArcticLink antifuse bits added here.

Table 8-15: Capability Register Settings (Offset 40h)

Bit(s)	Signal Name	Type	Reset Value	Function
31:27				Reserved.
26:24	VOLTAGE_SUPPORT	R	From Antifuse AF_VOLT_SUP	Voltage Support: Set Bit of AF_VOLT_SUP 2 to show support for 1.8V 1 to show support for 3.0V 0 to show support for 3.3V Recommended value to set is 001.
23	SUSPEND_SUPPORT	R	0	Suspend/Resume Support 0 = Not Supported

© 2007 QuickLogic Corporation

Table 8-15: Capability Register Settings (Offset 40h) (Continued)

Bit(s)	Signal Name	Type	Reset Value	Function
22	DMA_SUPPORT	R	From Antifuse AF_DMA_SUP	DMA Support 1= Supported 0 = Not supported, Should set to 0 unless a DMA is built into the fabric.
21	HIGH_SPEED_SUPPORT	R	1	High Speed Support 1= Supported
20:18	–			Reserved
17:16	MAX_BLK_LEN	R	From Antifuse AF_MAX_BLK_LEN	Max Block Length, Must set to 00 00 = 512 Bytes 01 = 1024 Bytes 10 = 2048 Bytes 11 = Reserved
15:14	–			Reserved
13:8	BASE_CLK_FREQ	R	From Antifuse AF_SDCLK_FREQ	Base Clock Frequency for SD
7	TIMEOUT_CLK_UNIT	R	From Antifuse AF_TIMEOUT_UNIT	Timeout Clock Unit 0 = KHz 1 = MHz
6	–			Reserved
5:0	TIMEOUT_CLK_FREQ	R	From Antifuse AF_TIMEOUT_FREQ	Timeout Clock Frequency

8.2.8 Maximum Current Capabilities Register (Offset 48h)

This register is defined in the SD Host Controller Specification. **Table 8-16** provides the ArcticLink antifuse bits added here.

Table 8-16: Maximum Current Capabilities Register (Offset 048h)

Bit	Signal Name	Type	Reset Value	Function
31:24	–			Reserved
23:16	MAX_CUR_18V	R	From Antifuse AF_MAX_CUR_18V	Maximum current for 1.8V
15:8	MAX_CUR_30V	R	From Antifuse AF_MAX_CUR_30V	Maximum current for 3.0V
7:0	MAX_CUR_33V	R	From Antifuse AF_MAX_CUR_33V	Maximum current for 3.3V

© 2007 QuickLogic Corporation

8.2.9 CE-ATA Control Register (Offset 100h)

The CE-ATA Register Block will have a Control Register at offset 0x100 from the CE-ATA base offset in the memory map.

Table 8-17: CE-ATA Control Register

Bit	Signal Name	Type	Reset	Function
31:10				Reserved
9	DREQ_EN	RW	0	PXA DMA mode support 1= Enable use of DREQ signal 0 = Disable use of DREQ signal
8	FIFO_EN	RW	1	Enable Dynamic Buffer Management mode in order to increase data throughput Dynamic Buffer Management support. 1= Enable use of dynamic buffering mode 0 = Disable use of dynamic buffering mode
7:5				Reserved
4	CEATA_MODE_EN	RW	0	CEATA Mode support 1= Enable use of CEATA mode 0 = Disable use of CEATA mode
3:2				Reserved
1	CEATA_CCSD_EN	RW	0	CE-ATA Command Complete Signal Disable Issuing (ignored if bit [4]=0) 1= Enable issuing the Signal 0 = Disable issuing the Signal
0	CEATA_CCS_INT_EN	RW	0	CE-ATA Command Complete Signal Interrupt Enable (ignored if bit [4]=0) 1= Enable CEATA CCS Interrupt 0 = Disable CEATA CSS Interrupt

8.3 USB OTG Control and Status Registers

8.3.1 USB OTG Control and Status Overview

Applications control the Hi-Speed USB 2.0 OTG core by reading from and writing to the Control and Status Registers (CSRs) through the Internal Bus Slave interface. CSR registers are 32 bits wide, and their addresses are 32-bit block aligned.

There are 5 classifications of CSRs as follows.

- Core Global Registers
- Host Mode Registers
  - Host Global Registers
  - Host Port CSRs
  - Host Channel-Specific Registers
- Device Mode Registers

© 2007 QuickLogic Corporation

  - Device Global Registers
  - Device Endpoint-Specific Registers
- Power and Clock-Gating Registers
- Data FIFO (DFIFO) Access Registers

Only the Core Global, Power and Clock Gating, Data FIFO Access, and Host Port registers can be accessed in both Host and Device modes. When the Hi-Speed USB 2.0 OTG core is operating in one mode, either Device or Host, the application must not access registers from the other mode. If an illegal access occurs, a Mode Mismatch interrupt is generated and reflected in the Core Interrupt register (GINTSTS.ModeMis).

When the core switches from one mode to another, the registers in the new mode of operation must be reprogrammed as they would be after a power-on reset.

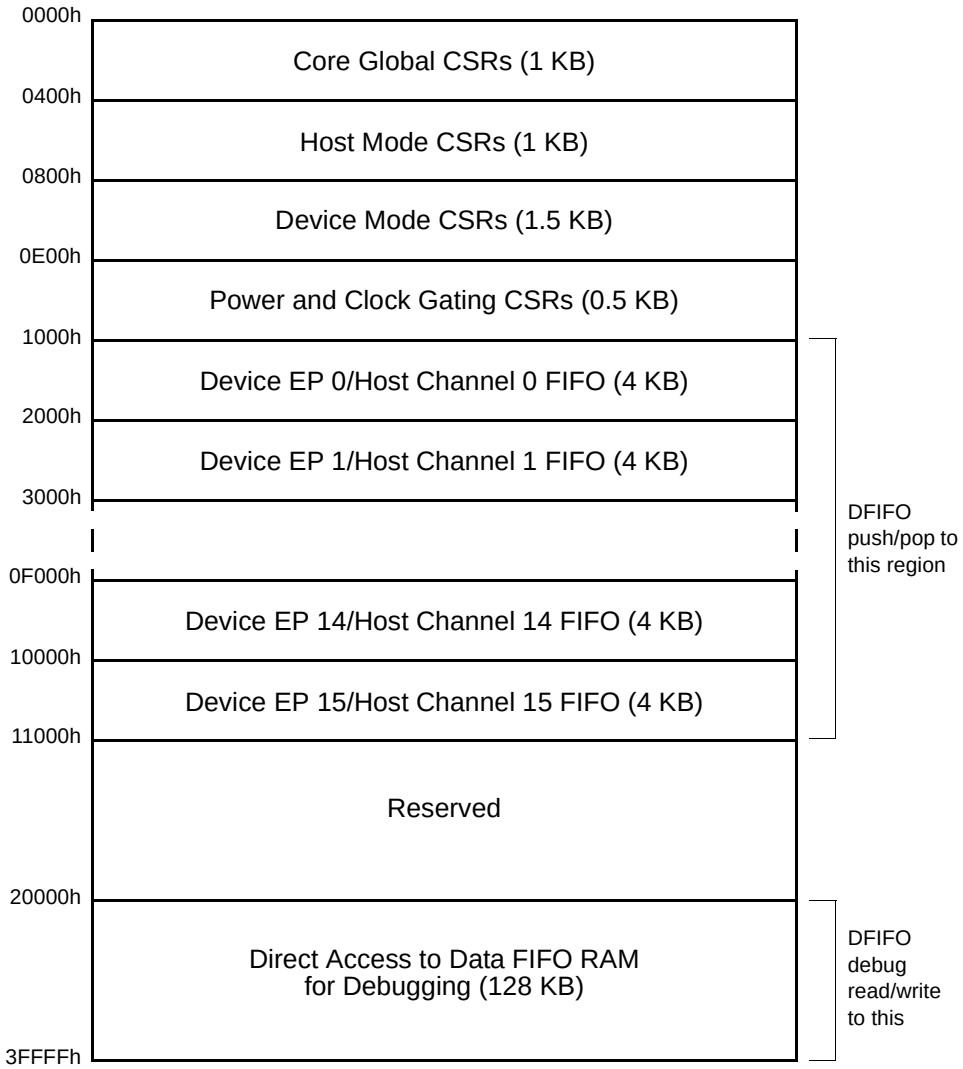
8.3.2 CSR Memory Map

Figure 8-2 shows the CSR address map. This map is fixed and does not depend on the core's configuration (for example, how many endpoints are implemented). Host and Device mode registers occupy different addresses. All registers are implemented in the Internal Bus Clock domain.

NOTE: The Hi-Speed USB 2.0 OTG Core Register block will have an OTG Control and Status register at offset 40000h in the ArcticLink's recommended memory map, **Figure 8-1**

© 2007 QuickLogic Corporation

Figure 8-2: OTG CSR Memory Map



8.3.2.1 Register Maps

The tables in this section provide high-level summaries of each register and register group.

Table 8-18: Summary of Register and Register Group

Register Name	Name of register types and register names ordered by offset address
Acronym	Shorthand names for registers that are mapped to the offset address. The first letter is a prefix for the register type: G: Core Global H: Host mode D: Device mode
Offset Address	Address, in hexadecimal (h), of the first byte of each register.

NOTE: FIFO size and FIFO depth are used interchangeably.

8.3.2.2 Core Global CSR Map

These registers are available in both Host and Device modes.

Table 8-19: Core Global CSR Map

Register Name	Acronym	Offset Address	Page
Core Global Registers		**000h–3FFh**	86
OTG Control and Status Register	GOTGCTL	000h	86
OTG Interrupt Register	GOTGINT	004h	86
Core Internal Bus Configuration Register	GAHBCFG	008h	88
Core USB Configuration Register	GUSBCFG	00Ch	90
Core Reset Register	GRSTCTL	010h	93
Core Interrupt Register	GINTSTS	014h	96
Core Interrupt Mask Register	GINTMSK	018h	100
Receive Status Debug Read Register (Read Only)	GRXSTSR	01Ch	102
Receive Status Read/Pop Register (Read Only)	GRXSTSP	020h	102
Receive FIFO Size Register	GRXFSIZ	024h	103
Non-periodic Transmit FIFO Size Register	GNPTXFSIZ	028h	103
Non-periodic Transmit FIFO/Queue Status Register (Read Only)	GNPTXSTS	02Ch	103
I2C Access Register	GI2CCTL	030h	105
PHY Vendor Control Register	GPVNDCTL	034h	107
General Purpose Input/Output Register	GGPIO	038h	107
User ID Register	GUID	03Ch	108
Reserved		040h	
User HW Config1 Register (Read Only)	GHWCFG1	044h	108
User HW Config2 Register (Read Only)	GHWCFG2	048h	108
User HW Config3 Register (Read Only)	GHWCFG3	04Ch	109
User HW Config4 Register (Read Only)	GHWCFG4	050h	110
Reserved		054h–0FFh	

© 2007 QuickLogic Corporation

Table 8-19: Core Global CSR Map (Continued)

Register Name	Acronym	Offset Address	Page
Host Periodic Transmit FIFO Size Register	HPTXFSIZ	100h	111
Device Periodic Transmit FIFO-n Size Register	DPTXFSIZn	104h–13Ch	112
Reserved		140h–3FFh	

8.3.2.3 Host Mode CSR Map

These registers must be programmed every time the core changes to Host mode.

Table 8-20: Host Mode CSR Map

Register Name	Acronym	Offset Address	Page
Host Global Registers		**400h–7FFh**	114
Host Configuration Register	HCFG	400h	114
Host Frame Interval Register	HFIR	404h	114
Host Frame Number/Frame Time Remaining Register	HFNUM	408h	115
Reserved		40Ch	
Host Periodic Transmit FIFO/Queue Status Register	HPTXSTS	410h	116
Host All Channels Interrupt Register	HAINT	414h	117
Host All Channels Interrupt Mask Register	HAINTMSK	418h	117
Host Port Control and Status Register		**440h–47Ch**	117
Host Port Control and Status Register	HPRT	440h	117
Reserved		444h–4FCh	
Host Channel-Specific Registers (n = 0 to 15)		**500h–6FCh**	120
Host Channel 0 Characteristics Register	HCCHARn	500h	120
Host Channel 0 Split Control Register	HCSPLTn	504h	121
Host Channel 0 Interrupt Register	HCINTn	508h	121
Host Channel 0 Interrupt Mask Register	HCINTMSKn	50Ch	122
Host Channel 0 Transfer Size Register	HCTSIZn	510h	123
Host Channel 0 DMA Address Register	HCDMAn	514h	123
Reserved		518h–51Ch	
Host Channel 1 Registers		520h–53Ch	120
Host Channel 2 Registers		540h–55Ch	120
...		...	
Host Channel 14 Registers		6C0h–6DCh	120
Host Channel 15 Registers		6E0h–6FCh	120
Reserved		6FDh–7FFh	

© 2007 QuickLogic Corporation

8.3.2.4 Device Mode CSR Map

These registers must be programmed every time the core changes to Device mode.

Table 8-21: Device Mode CSR Map

Register Name	Acronym	Offset Address	Page
Device Global Registers		**800h–BFFh**	125
Device Configuration Register	DCFG	800h	125
Device Control Register	DCTL	804h	126
Device Status Register (Read Only)	DSTS	808h	128
Reserved		80Ch	
Device IN Endpoint Common Interrupt Mask Register	DIEPMSK	810h	129
Device OUT Endpoint Common Interrupt Mask Register	DOEPMSK	814h	129
Device All Endpoints Interrupt Register	DAINT	818h	130
Device All Endpoints Interrupt Mask Register	DAINTMSK	81Ch	130
Device IN Token Sequence Learning Queue Read Register 1 (Read Only)	DTKNQR1	820h	131
Device IN Token Sequence Learning Queue Read Register 2 (Read Only)	DTKNQR2	824h	131
Device IN Token Sequence Learning Queue Read Register 3 (Read Only)	DTKNQR3	830h	131
Device IN Token Sequence Learning Queue Read Register 4 (Read Only)	DTKNQR4	834h	132
Device VBUS Discharge Time Register	DVBUSDIS	828h	132
Device VBUS Pulsing Time Register	DVBUSPULSE	82Ch	132
Reserved		830h	
Device IN Endpoint FIFO Empty Interrupt Mask Register	DIEPEMPMSK	834h	133
Reserved		838h–8FFh	
Device Control IN Endpoint 0 Control Register	DIEPCTLn	900h	133
Reserved		904h	
Device IN Endpoint 0 Interrupt Register	DIEPINTn	908h	140
Reserved		90Ch	
Device IN Endpoint 0 Transfer Size Register	DIEPTSIZn	910h	141
Device IN Endpoint 0 DMA Address Register	DIEPDMAn	914h	144
Device IN Endpoint Transmit FIFO Status Register	DTXFSTSn	918h	145
Reserved		918h–91Ch	
Device IN Endpoint 1 Registers		920h–93Ch	136
Device IN Endpoint 2 Registers		940h–95Ch	136
...		...	
Device IN Endpoint 14 Registers		AC0h–ADCh	136
Device IN Endpoint 15 Registers		AE0h–AFCh	136

© 2007 QuickLogic Corporation

Table 8-21: Device Mode CSR Map (Continued)

Register Name	Acronym	Offset Address	Page
Device Logical OUT Endpoint-Specific Registers		**B00h–CFCh**	134
Device Control OUT Endpoint 0 Control Register	DOEPCTLn	B00h	134
Reserved		B04h	
Device OUT Endpoint 0 Interrupt Register	DOEPINTn	B08h	140
Reserved		B0Ch	
Device OUT Endpoint 0 Transfer Size Register	DOEPTSIZn	B10h	141
Device OUT Endpoint 0 DMA Address Register	DOEPDMAn	B14h	144
Reserved		B14h–B1Ch	
Device OUT Endpoint 1 Registers		B20h–B3Ch	136
Device OUT Endpoint 2 Registers		B40h–B5Ch	136
...		...	
Device OUT Endpoint 14 Registers		CC0h–CDCh	136
Device OUT Endpoint 15 Registers		CE0h–CFCh	136
Reserved		CFDh–DFFh	

8.3.2.5 Power and Clock Gating CSR Map

There is a single register for power and clock gating. It is available in both Host and Device modes.

Table 8-22: Power and Clock Gating Register

Register Name	Acronym	Offset Address	Page
Power and Clock Gating Registers		**E00h–FFFh**	145
Power and Clock Gating Control Register	PCGCR	E00h	145
Reserved		E05h–FFFh	

8.3.2.6 Data FIFO (DFIFO) Access Register Map

These registers, available in both Host and Device modes, are used to read or write the FIFO space for a specific endpoint or a channel, in a given direction. If a host channel is of type IN, the FIFO can only be read on the channel. Similarly, if a host channel is of type OUT, the FIFO can only be written on the channel.

Table 8-23: Data FIFO (DFIFO) Access Register Map

FIFO Access Register Section	Address Range	Access
Device IN Endpoint 0/Host OUT Channel 0: DFIFO Write Access Device OUT Endpoint 0/Host IN Channel 0: DFIFO Read Access	1000h–1FFCh	WO/RO
Device IN Endpoint 1/Host OUT Channel 1: DFIFO Write Access Device OUT Endpoint 1/Host IN Channel 1: DFIFO Read Access	2000h–2FFCh	WO/RO

© 2007 QuickLogic Corporation

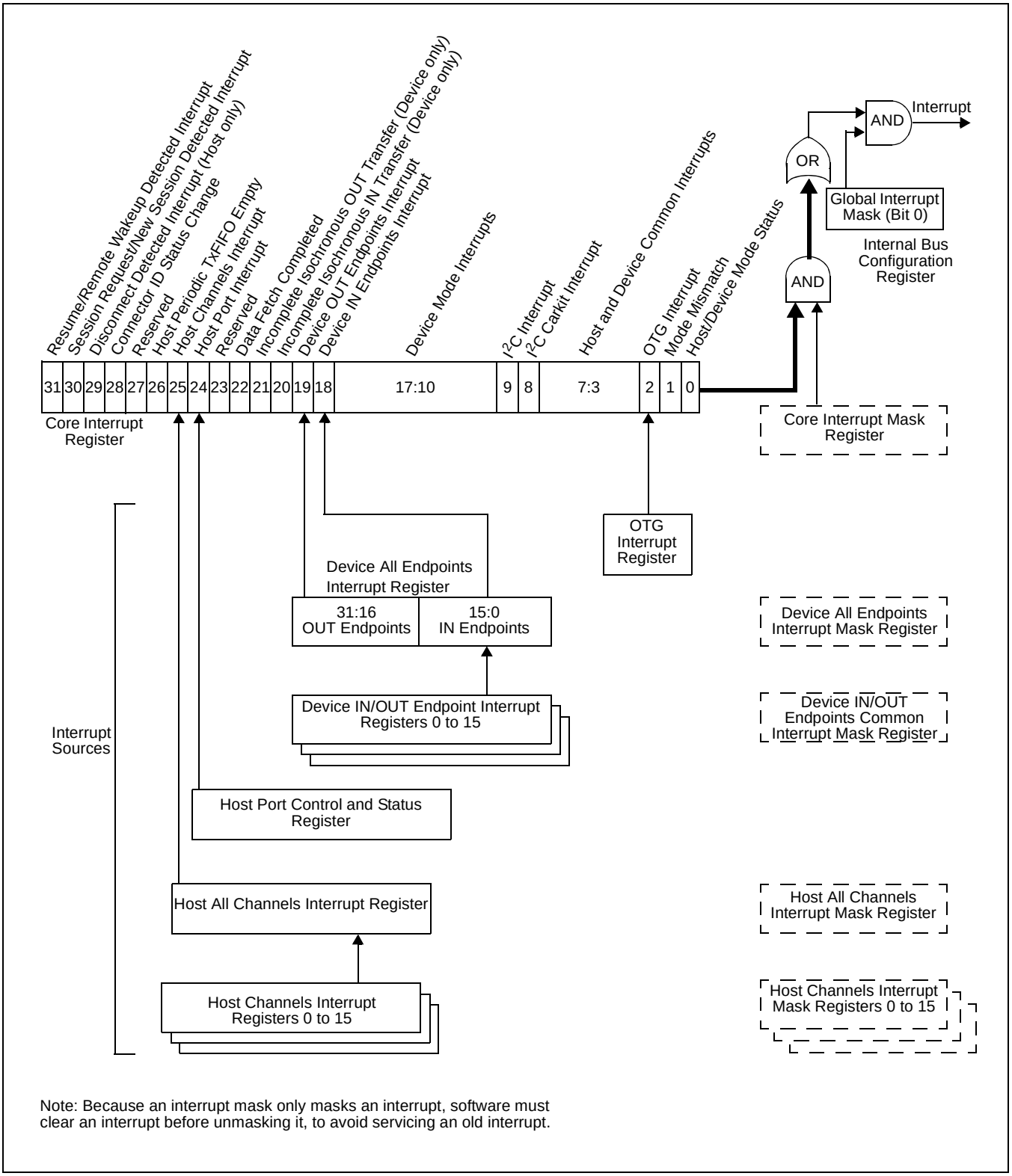
Table 8-23: Data FIFO (DFIFO) Access Register Map (Continued)

FIFO Access Register Section	Address Range	Access
...	...	...
Device IN Endpoint 14/Host OUT Channel 14: DFIFO Write Access Device OUT Endpoint 14/Host IN Channel 14: DFIFO Read Access	F000h–FFFCh	WO/RO
Device IN Endpoint 15/Host OUT Channel 15: DFIFO Write Access Device OUT Endpoint 15/Host IN Channel 15: DFIFO Read Access	10000h–10FFCh	WO/RO

© 2007 QuickLogic Corporation

8.3.2.7 Interrupt Hierarchy

Figure 8-3: Interrupt Hierarchy


Resume/Remote Wakeup Detected Interrupt
Session Request/New Session Detected Interrupt
Disconnect Detected Interrupt (Host only)
Connector ID Status Change
Reserved
Host Periodic TxFIFO Empty
Host Channels Interrupt
Host Port Interrupt
Reserved
Data Fetch Completed
Incomplete Isochronous OUT Transfer (Device only)
Incomplete Isochronous IN Transfer (Device only)
Device OUT Endpoints Interrupt
Device IN Endpoints Interrupt
Device Mode Interrupts
I2C Interrupt
I2C Carkit Interrupt
Host and Device Common Interrupts
OTG Interrupt
Mode Mismatch
Host/Device Mode Status
AND
Interrupt
OR
Global Interrupt Mask (Bit 0)
Internal Bus Configuration Register
AND
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17:10 9 8 7:3 2 1 0
Core Interrupt Register
Core Interrupt Mask Register
OTG Interrupt Register
Device All Endpoints Interrupt Register
31:16 OUT Endpoints
15:0 IN Endpoints
Device All Endpoints Interrupt Mask Register
Device IN/OUT Endpoint Interrupt Registers 0 to 15
Device IN/OUT Endpoints Common Interrupt Mask Register
Interrupt Sources
Host Port Control and Status Register
Host All Channels Interrupt Register
Host All Channels Interrupt Mask Register
Host Channels Interrupt Registers 0 to 15
Host Channels Interrupt Mask Registers 0 to 15

Note: Because an interrupt mask only masks an interrupt, software must clear an interrupt before unmasking it, to avoid servicing an old interrupt.

8.3.3 Register Descriptions

8.3.3.1 Core Global Registers

These registers are available in both Host and Device modes, and do not need to be reprogrammed when switching between these modes.

8.3.3.1.1 OTG Control and Status Register (GOTGCTL)

Offset: 000h

The OTG Control and Status register controls the behavior and reflects the status of the OTG function of the core.

Table 8-24: OTG Control and Status Register: GOTGCTL

Field	Description	Mode	Reset	Access
31:20	Reserved		12'h0	
19	B-Session Valid (BSesVld) Indicates the Device mode transceiver status. 1'b0: B-session is not valid. 1'b1: B-session is valid.	Device only	1'b0	RO
18	A-Session Valid (ASesVld) Indicates the Host mode transceiver status. 1'b0: A-session is not valid 1'b1: A-session is valid	Host only	1'b0	RO
17	Long/Short Debounce Time (DbncTime) Indicates the debounce time of a detected connection. 1'b0: Long debounce time, used for physical connections (100 ms + 2.5 µs) 1'b1: Short debounce time, used for soft connections (2.5 µs)	Host only	1'b0	RO
16	Connector ID Status (ConIDSts) Indicates the connector ID status on a connect event. 1'b0: The USB OTG core is in A-Device mode 1'b1: The USB OTG core is in B-Device mode	Host and Device	1'b1	RO
15:12	Reserved		4'h0	
11	Device HNP Enabled (DevHNPEn) The application sets this bit when it successfully receives a SetFeature.SetHNPEnable command from the connected USB host. 1'b0: HNP is not enabled in the application 1'b1: HNP is enabled in the application	Device only	1'b0	R_W
10	Host Set HNP Enable (HstSetHNPEn) The application sets this bit when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) on the connected device. 1'b0: Host Set HNP is not enabled 1'b1: Host Set HNP is enabled	Host only	1'b0	R_W

© 2007 QuickLogic Corporation

Table 8-24: OTG Control and Status Register: GOTGCTL (Continued)

Field	Description	Mode	Reset	Access
9	HNP Request (HNPReq) The application sets this bit to initiate an HNP request to the connected USB host. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. 1'b0: No HNP request 1'b1: HNP request	Device only	1'b0	R_W
8	Host Negotiation Success (HstNegScs) The core sets this bit when host negotiation is successful. The core clears this bit when the HNP Request (HNPReq) bit in this register is set. 1'b0: Host negotiation failure 1'b1: Host negotiation success	Device only	1'b0	RO
7:2	Reserved		4'h0	
1	Session Request (SesReq) The application sets this bit to initiate a session request on the USB. The application can clear this bit by writing a 0 when the Host Negotiation Success Status Change bit in the OTG Interrupt register (GOTGINT.HstNegSucStsChng) is set. The core clears this bit when the HstNegSucStsChng bit is cleared. If you use the USB 1.1 Full-Speed Serial Transceiver interface to initiate the session request, the application must wait until the VBUS discharges to 0.2 V, after the B-Session Valid bit in this register (GOTGCTL.BSesVld) is cleared. This discharge time varies between different PHYs and can be obtained from the PHY vendor. 1'b0: No session request 1'b1: Session request	Device only	1'b0	R_W
0	Session Request Success (SesReqScs) The core sets this bit when a session request initiation is successful. 1'b0: Session request failure 1'b1: Session request success	Device only	1'b0	RO

Offset: 004h

The application reads this register whenever there is an OTG interrupt and clears the bits in this register to clear the OTG interrupt. It is shown in **Figure 8-3**.

Table 8-25: OTG Interrupt Register: GOTGINT

Field	Description	Mode	Reset	Access
31:20	Reserved		12'h0	
19	Debounce Done (DbnceDone) The core sets this bit when the debounce is completed after the device connect. The application can start driving USB reset after seeing this interrupt. This bit is only valid when the HNP Capable or SRP Capable bit is set in the Core USB Configuration register (GUSBCFG.HNPCap or GUSBCFG.SRPCap, respectively).	Host and Device	1'b0	R_SS_WC
18	A-Device Timeout Change (ADevTOUTChg) The core sets this bit to indicate that the A-device has timed out while waiting for the B-device to connect.	Host and Device	1'b0	R_SS_WC
17	Host Negotiation Detected (HstNegDet) The core sets this bit when it detects a host negotiation request on the USB.	Host and Device	1'b0	R_SS_WC

Table 8-25: OTG Interrupt Register: GOTGINT (Continued)

Field	Description	Mode	Reset	Access
16:10	Reserved		6'h0	
9	Host Negotiation Success Status Change (HstNegSucStsChng) The core sets this bit on the success or failure of a USB host negotiation request. The application must read the Host Negotiation Success bit of the OTG Control and Status register (GOTGCTL.HstNegScs) to check for success or failure.	Host and Device	1'b0	R_SS_WC
8	Session Request Success Status Change (SesReqSucStsChng) The core sets this bit on the success or failure of a session request. The application must read the Session Request Success bit in the OTG Control and Status register (GOTGCTL.SesReqScs) to check for success or failure.	Host and Device	1'b0	R_SS_WC
7:3	Reserved		6'h0	
2	Session End Detected (SesEndDet) The core sets this bit when the utmiotg_bvalid signal is deasserted.	Host and Device	1'b0	R_SS_WC
1:0	Reserved		2'h0	

8.3.3.1.2 Core Internal Bus Configuration Register (GAHBCFG)

Offset: 008h

This register can be used to configure the core after power-on or a change in mode of operation. This register mainly contains Internal Bus system-related configuration parameters. Do not change this register after the initial programming. The application must program this register before starting any transactions on either the Internal Bus or the USB.

© 2007 QuickLogic Corporation

Table 8-26: Core Internal Bus Configuration Register: GAHBCFG

Field	Description	Mode	Reset	Access
31:9	Reserved		22'h0	
8	Periodic TxFIFO Empty Level (PTxFEmpLvl) Indicates when the Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.PTxFEmp) is triggered. This bit is used only in Slave mode. 1'b0: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is half empty 1'b1: GINTSTS.PTxFEmp interrupt indicates that the Periodic TxFIFO is completely empty	Host only	1'b0	R_W
7	Non-Periodic TxFIFO Empty Level (NPTxFEmpLvl) This bit is used only in Slave mode. In host mode and with Shared FIFO with device mode, this bit indicates when the Non-Periodic TxFIFO Empty Interrupt bit in the Core Interrupt register (GINTSTS.NPTxFEmp) is triggered. With dedicated FIFO in device mode, this bit indicates when IN endpoint Transmit FIFO empty interrupt (DIEPINTn.TxFEmp) is triggered. Host mode and with Shared FIFO with device mode:- 1'b0: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is half empty 1'b1: GINTSTS.NPTxFEmp interrupt indicates that the Non-Periodic TxFIFO is completely empty Dedicated FIFO in device mode :- 1'b0: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is half empty 1'b1: DIEPINTn.TxFEmp interrupt indicates that the IN Endpoint TxFIFO is completely empty	Host and Device	1'b0	R_W
6	Reserved		1'b0	

© 2007 QuickLogic Corporation

Table 8-26: Core Internal Bus Configuration Register: GAHBCFG (Continued)

Field	Description	Mode	Reset	Access
5	DMA Enable (DMAEn) 1'b0: Core operates in Slave mode 1'b1: Core operates in a DMA mode	Host and Device	1'b0	R_W
4:1	Burst Length/Type (HBstLen) This field is used in both External and Internal DMA modes. In External DMA mode, these bits appear on dma_burst[3:0] ports. External DMA Mode—defines the DMA burst length in terms of 32-bit words: 4'b0000: 1 word 4'b0001: 4 words 4'b0010: 8 words 4'b0011: 16 words 4'b0100: 32 words 4'b0101: 64 words 4'b0110: 128 words 4'b0111: 256 words Others: Reserved Internal DMA Mode—Internal Bus Master burst type: 4'b0000 Single 4'b0001 INCR 4'b0011 INCR4 4'b0101 INCR8 4'b0111 INCR16 Others: Reserved	Host and Device	4'b0	R_W
0	Global Interrupt Mask (GlblIntrMsk) The application uses this bit to mask or unmask the interrupt line assertion to itself. Irrespective of this bit's setting, the interrupt status registers are updated by the core. 1'b0: Mask the interrupt assertion to the application. 1'b1: Unmask the interrupt assertion to the application.	Host and Device	1'b0	R_W

8.3.3.1.3 Core USB Configuration Register (GUSBCFG)

Offset: 00Ch

This register can be used to configure the core after power-on or a changing to Host mode or Device mode. It contains USB and USB-PHY related configuration parameters. The application must program this register before starting any transactions on either the Internal Bus or the USB. Do not make changes to this register after the initial programming.

© 2007 QuickLogic Corporation

Table 8-27: Core USB Configuration Register: GUSBCFG

Field	Description	Mode	Reset	Access
31	Corrupt Tx packet This bit is for debug purposes only. Never set this bit to 1.	Host and Device	1'b0	R_W
30	Force Device Mode (ForceDevMode) Writing a 1 to this bit will force the core to device mode irrespective of utmiotg_iddig input pin. 1'b0 : Normal Mode. 1'b1 : Force Device Mode.	Host and Device	1'b0	R_W
29	Force Host Mode (ForceHstMode) Writing a 1 to this bit will force the core to host mode irrespective of utmiotg_iddig input pin. 1'b0 : Normal Mode. 1'b1 : Force Host Mode.	Host and Device	1'b0	R_W
28:23	Reserved		6'h0	
22	TermSel DLine Pulsing Selection (TermSelDLPulse) This bit selects utmi_termselect to drive data line pulse during SRP. 1'b0: Data line pulsing using utmi_txvalid (default). 1'b1: Data line pulsing using utmi_termsel.	Device Only	1'b0	R_W
21	ULPI External VBUS Indicator (ULPIExtVbusIndicator) This bit indicates to the ULPI PHY to use an external VBUS over-current indicator. 1'b0: PHY uses internal VBUS valid comparator. 1'b1: PHY uses external VBUS valid comparator.	Host Only	1'b0	R_W
20	ULPI External VBUS Drive (ULPIExtVbusDrv) This bit selects between internal or external supply to drive 5V on VBUS, in ULPI PHY. 1'b0: PHY drives VBUS using internal charge pump (default). 1'b1: PHY drives VBUS using external supply.	Host Only	1'b0	R_W
19	ULPI Clock SuspendM (ULPIClkSusM) This bit sets the ClockSuspendM bit in the Interface Control register on the ULPI PHY. This bit applies only in serial or carkit modes. 1'b0: PHY powers down internal clock during suspend. 1'b1: PHY does not power down internal clock.	Host and Device	1'b0	R_W
18	ULPI Auto Resume (ULPIAutoRes) This bit sets the AutoResume bit in the Interface Control register on the ULPI PHY. 1'b0: PHY does not use AutoResume feature. 1'b1: PHY uses AutoResume feature.	Host and Device	1'b0	R_W
17	ULPI FS/LS Select (ULPIFsLs) The application uses this bit to select the FS/LS serial interface for the ULPI PHY. This bit is valid only when the FS serial transceiver is selected on the ULPI PHY. 1'b0: ULPI interface 1'b1: ULPI FS/LS serial interface	Host and Device	1'b0	R_W
16	UTMIFS or i2c compatible serial bus Interface Select (OtgI2CSel) The application uses this bit to select the i2c compatible serial bus interface. 1'b0: UTMI USB 1.1 Full-Speed interface for OTG signals 1'b1: i2c compatible serial bus interface for OTG signals	Host and Device	1'b0	RO / R_W

© 2007 QuickLogic Corporation

Table 8-27: Core USB Configuration Register: GUSBCFG (Continued)

Field	Description	Mode	Reset	Access
15	PHY Low-Power Clock Select (PhyLPwrClkSel) Selects either 480-MHz or 48-MHz (low-power) PHY mode. In FS and LS modes, the PHY can usually operate on a 48-MHz clock to save power. 1'b0: 480-MHz Internal PLL clock 1'b1: 48-MHz External Clock In 480 MHz mode, the UTMI interface operates at either 60 or 30-MHz, depending upon whether 8- or 16-bit data width is selected. In 48-MHz mode, the UTMI interface operates at 48 MHz in FS mode and at either 48 or 6 MHz in LS mode (depending on the PHY vendor). This bit drives the utmi_fsls_low_power core output signal, and is valid only for UTMI+ PHYs.	Host and Device	1'b0	R_W
14	Reserved		1'b0	
13:10	USB Turnaround Time (USBTrdTim) Sets the turnaround time in PHY clocks. Specifies the response time for a MAC request to the Packet FIFO Controller (PFC) to fetch data from the DFIFO (SPRAM). This must be programmed to 4'h5: When the MAC interface is 16-bit UTMI+ . 4'h9: When the MAC interface is 8-bit UTMI+ . **Note:** The values above are calculated for the minimum Internal Bus frequency of 30 MHz. USB turnaround time is critical for certification where long cables and 5-Hubs are used, so if you need the Internal Bus to run at less than 30 MHz, and if USB turnaround time is not critical, these bits can be programmed to a larger value.	Device only	4'h5	R_W
9	HNP-Capable (HNPCap) The application uses this bit to control the USB OTG core's HNP capabilities. 1'b0: HNP capability is not enabled. 1'b1: HNP capability is enabled.	Host and Device	1'b0	RO / R_W
8	SRP-Capable (SRPCap) The application uses this bit to control the USB OTG core's SRP capabilities. If the core operates as a non-SRP-capable B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. 1'b0: SRP capability is not enabled. 1'b1: SRP capability is enabled.	Host and Device	1'b1	RO / R_W
7	ULPI DDR Select (DDRSel) The application uses this bit to select a Single Data Rate (SDR) or Double Data Rate (DDR) or ULPI interface. 1'b0: Single Data Rate ULPI Interface, with 8-bit-wide data bus 1'b1: Double Data Rate ULPI Interface, with 4-bit-wide data bus	Host and Device	1'b0	R_W
6	USB 2.0 High-Speed PHY or USB 1.1 Full-Speed Serial Transceiver Select (PHYSel) The application uses this bit to select either a high-speed UTMI+ or ULPI PHY, or a full-speed transceiver. 1'b0: USB 2.0 high-speed UTMI+ or ULPI PHY 1'b1: USB 1.1 full-speed serial transceiver	Host and Device	1'b0	WO / R_W

© 2007 QuickLogic Corporation

Table 8-27: Core USB Configuration Register: GUSBCFG (Continued)

Field	Description	Mode	Reset	Access
5	Full-Speed Serial Interface Select (FSIntf) The application uses this bit to select either a unidirectional or bidirectional USB 1.1 full-speed serial transceiver interface. 1'b0: 6-pin unidirectional full-speed serial interface 1'b1: 3-pin bidirectional full-speed serial interface	Host and Device	1'b0	WO /R_W
4	ULPI or UTMI+ Select (ULPI_UTMI_Sel) The application uses this bit to select either a UTMI+ interface or ULPI Interface. 1'b0: UTMI+ Interface 1'b1: ULPI Interface	Host and Device	1'b0	RO / R_W
3	PHY Interface (PHYIf) The application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface. When a ULPI PHY is chosen, this must be set to 8-bit mode. 1'b0: 8 bits 1'b1: 16 bits	Host and Device	1'b0	RO / R_W
2:0	HS/FS Timeout Calibration (TOutCal) The number of PHY clocks that the application programs in this field is added to the high-speed/full-speed interpacket timeout duration in the core to account for any additional delays introduced by the PHY. This can be required, because the delay introduced by the PHY in generating the linestate condition can vary from one PHY to another. The USB standard timeout value for high-speed operation is 736 to 816 (inclusive) bit times. The USB standard timeout value for full-speed operation is 16 to 18 (inclusive) bit times. The application must program this field based on the speed of enumeration. The number of bit times added per PHY clock are: High-speed operation: One 30-MHz PHY clock = 16 bit times One 60-MHz PHY clock = 8 bit times Full-speed operation: One 30-MHz PHY clock = 0.4 bit times One 60-MHz PHY clock = 0.2 bit times One 48-MHz PHY clock = 0.25 bit times	Host and Device	3'h0	R_W

8.3.3.1.4 Core Reset Register (GRSTCTL)

Offset: 010h

The application uses this register to reset various hardware features inside the core.

Table 8-28: Core Reset Register: GRSTCTL

Field	Description	Mode	Reset	Access
31	Internal Bus Master Idle (AHBIdle) Indicates that the Internal Bus Master State Machine is in the IDLE condition.	Host and Device	1'b1	RO
30	DMA Request Signal (DMAReq) Indicates that the DMA request is in progress. Used for debug.	Host and Device	1'b0	RO
29:11	Reserved		19'h0	

© 2007 QuickLogic Corporation

Table 8-28: Core Reset Register: GRSTCTL (Continued)

Field	Description	Mode	Reset	Access
10:6	TxFIFO Number (TxFNum) This is the FIFO number that must be flushed using the TxFIFO Flush bit. This field must not be changed until the core clears the TxFIFO Flush bit. 5'h0: Non-periodic TxFIFO flush in Host mode Non-periodic TxFIFO flush in device mode when in shared FIFO operation Tx FIFO 0 flush in device mode when in dedicated FIFO mode 5'h1: Periodic TxFIFO flush in Host mode Periodic TxFIFO 1 flush in Device mode when in shared FIFO operation TXFIFO 1 flush in device mode when in dedicated FIFO mode 5'h2: Periodic TxFIFO 2 flush in Device mode when in shared FIFO operation TXFIFO 2 flush in device mode when in dedicated FIFO mode ... 5'hF: Periodic TxFIFO 15 flush in Device mode when in shared FIFO operation TXFIFO 15 flush in device mode when in dedicated FIFO mode 5'h10: Flush all the transmit FIFOs in device or host mode.	Host and Device	5'h0	R_W
5	TxFIFO Flush (TxFFlsh) This bit selectively flushes a single or all transmit FIFOs, but cannot do so if the core is in the midst of a transaction. The application must write this bit only after checking that the core is neither writing to the TxFIFO nor reading from the TxFIFO. Verify using these registers: Read—NAK Effective Interrupt ensures the core is not reading from the FIFO Write—GRSTCTL.AHBIdle ensures the core is not writing anything to the FIFO. Flushing is normally recommended when FIFOs are reconfigured or when switching between Shared FIFO and Dedicated Transmit FIFO operation. FIFO flushing is also recommended during device endpoint disable. The application must wait until the core clears this bit before performing any operations. This bit takes eight clocks to clear, using the slower clock of phy_clk or hclk.	Host and Device	1'b0	R_WS_SC
4	RxFIFO Flush (RxFFlsh) The application can flush the entire RxFIFO using this bit, but must first ensure that the core is not in the middle of a transaction. The application must only write to this bit after checking that the core is neither reading from the RxFIFO nor writing to the RxFIFO. The application must wait until the bit is cleared before performing any other operations. This bit requires 8 clocks (slowest of PHY or Internal Bus clock) to clear.	Host and Device	1'b0	R_WS_SC
3	IN Token Sequence Learning Queue Flush (INTknQFlsh) This bit is valid only if it is = 0. The application writes this bit to flush the IN Token Sequence Learning Queue.	Device only	1'b0	R_WS_SC

© 2007 QuickLogic Corporation

Table 8-28: Core Reset Register: GRSTCTL (Continued)

Field	Description	Mode	Reset	Access
2	Host Frame Counter Reset (FrmCntrRst) The application writes this bit to reset the (micro)frame number counter inside the core. When the (micro)frame counter is reset, the subsequent SOF sent out by the core has a (micro)frame number of 0.	Host only	1'b0	R_WS_SC
1	HClk Soft Reset (HSftRst) The application uses this bit to flush the control logic in the Internal Bus Clock domain. Only Internal Bus Clock Domain pipelines are reset. FIFOs are not flushed with this bit. All state machines in the Internal Bus clock domain are reset to the Idle state after terminating the transactions on the Internal Bus, following the protocol. CSR control bits used by the Internal Bus clock domain state machines are cleared. To clear this interrupt, status mask bits that control the interrupt status and are generated by the Internal Bus clock domain state machine are cleared. Because interrupt status bits are not cleared, the application can get the status of any core events that occurred after it set this bit. This is a self-clearing bit that the core clears after all necessary logic is reset in the core. This can take several clocks, depending on the core's current state.	Host and Device	1'b0	R_WS_SC

Table 8-28: Core Reset Register: GRSTCTL (Continued)

Field	Description	Mode	Reset	Access
0	Core Soft Reset (CSftRst) Resets the hclk and phy_clock domains as follows: Clears the interrupts and all the CSR registers except the following register bits: PCGCCTL.RstPdwnModule PCGCCTL.GateHclk PCGCCTL.PwrClmp PCGCCTL.StopPPhyLPwrClkSelclk GUSBCFG.PhyLPwrClkSel GUSBCFG.DDRSel GUSBCFG.PHYSel GUSBCFG.FSIntf GUSBCFG.ULPI_UTMI_Sel GUSBCFG.PHYIf HCFG.FSLSPclkSel DCFG.DevSpd GGPIO All module state machines (except the Internal Bus Slave Unit) are reset to the IDLE state, and all the transmit FIFOs and the receive FIFO are flushed. Any transactions on the Internal Bus Master are terminated as soon as possible, after gracefully completing the last data phase of an Internal Bus transfer. Any transactions on the USB are terminated immediately. The application can write to this bit any time it wants to reset the core. This is a self-clearing bit and the core clears this bit after all the necessary logic is reset in the core, which can take several clocks, depending on the current state of the core. Once this bit is cleared software must wait at least 3 PHY clocks before doing any access to the PHY domain (synchronization delay). Software must also must check that bit 31 of this register is 1 (Internal Bus Master is IDLE) before starting any operation. Typically software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain has to be reset for proper operation.	Host and Device	1'b0	R_WS_SC

8.3.3.1.5 Core Interrupt Register (GINTSTS)

Offset: 014h

This register interrupts the application for system-level events in the current mode of operation (Device mode or Host mode). It is shown in Figure 8-3.

Some of the bits in this register are valid only in Host mode, while others are valid in Device mode only. This register also indicates the current mode of operation. In order to clear the interrupt status bits of type R_SS_WC, the application must write 1'b1 into the bit.

The FIFO status interrupts are read only; once software reads from or writes to the FIFO while servicing these interrupts, FIFO interrupt conditions are cleared automatically.

© 2007 QuickLogic Corporation

Table 8-29: Core Interrupt Register: GINTSTS

Field	Description	Mode	Reset	Access
31	Resume/Remote Wakeup Detected Interrupt (WkUpInt) In Device mode, this interrupt is asserted when a resume is detected on the USB. In Host mode, this interrupt is asserted when a remote wakeup is detected on the USB.	Host and Device	1'b0	R_SS_WC
30	Session Request/New Session Detected Interrupt (SessReqInt) In Host mode, this interrupt is asserted when a session request is detected from the device. In Device mode, this interrupt is asserted when the utmiotg_bvalid signal goes high.	Host and Device	1'b0	R_SS_WC
29	Disconnect Detected Interrupt (DisconnInt) Asserted when a device disconnect is detected.	Host and Device	1'b0	R_SS_WC
28	Connector ID Status Change (ConIDStsChng) The core sets this bit when there is a change in connector ID status.	Host and Device	1'b0	R_SS_WC
27	Reserved		1'b0	
26	Periodic TxFIFO Empty (PTxFEmp) Asserted when the Periodic Transmit FIFO is either half or completely empty and there is space for at least one entry to be written in the Periodic Request Queue. The half or completely empty status is determined by the Periodic TxFIFO Empty Level bit in the Core Internal Bus Configuration register (GAHBCFG.PTxFEmpLvl).	Host only	1'b0	RO
25	Host Channels Interrupt (HChInt) The core sets this bit to indicate that an interrupt is pending on one of the channels of the core (in Host mode). The application must read the Host All Channels Interrupt (HAINT) register to determine the exact number of the channel on which the interrupt occurred, and then read the corresponding Host Channel-n Interrupt (HCINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the HCINTn register to clear this bit.	Host only	1'b0	RO
24	Host Port Interrupt (PrtInt) The core sets this bit to indicate a change in port status of one of the USB OTG core ports in Host mode. The application must read the Host Port Control and Status (HPRT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the Host Port Control and Status register to clear this bit.	Host only	**1'b0**	**RO**
23	Reserved		**1'b0**	

© 2007 QuickLogic Corporation

Table 8-29: Core Interrupt Register: GINTSTS (Continued)

Field	Description	Mode	Reset	Access
22	Data Fetch Suspended (FetSusp) This interrupt is valid only in DMA mode. This interrupt indicates that the core has stopped fetching data for IN endpoints due to the unavailability of TxFIFO space or Request Queue space. This interrupt is used by the application for an endpoint mismatch algorithm. For example, after detecting an endpoint mismatch, the application: Sets a global non-periodic IN NAK handshake Disables In endpoints Flushes the FIFO Determines the token sequence from the IN Token Sequence Learning Queue Re-enables the endpoints Clears the global non-periodic IN NAK handshake If the global non-periodic IN NAK is cleared, the core has not yet fetched data for the IN endpoint, and the IN token is received: the core generates an “IN token received when FIFO empty” interrupt. The OTG then sends the host a NAK response. To avoid this scenario, the application can check the GINTSTS.FetSusp interrupt, which ensures that the FIFO is full before clearing a global NAK handshake. Alternatively, the application can mask the “IN token received when FIFO empty” interrupt when clearing a global IN NAK handshake.	Device only	1’b0	R_SS_WC
21	Incomplete Periodic Transfer (incomplP) In Host mode, the core sets this interrupt bit when there are incomplete periodic transactions still pending which are scheduled for the current microframe. Incomplete Isochronous OUT Transfer (incompISOOUT) The Device mode, the core sets this interrupt to indicate that there is at least one isochronous OUT endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	Host only Device only	1’b0	R_SS_WC
20	Incomplete Isochronous IN Transfer (incompISOIN) The core sets this interrupt to indicate that there is at least one isochronous IN endpoint on which the transfer is not completed in the current microframe. This interrupt is asserted along with the End of Periodic Frame Interrupt (EOPF) bit in this register.	Device only	1’b0	R_SS_WC
19	OUT Endpoints Interrupt (OEPInt) The core sets this bit to indicate that an interrupt is pending on one of the OUT endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the OUT endpoint on which the interrupt occurred, and then read the corresponding Device OUT Endpoint-n Interrupt (DOEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DOEPINTn register to clear this bit.	Device only	1’b0	RO
18	IN Endpoints Interrupt (IEPInt) The core sets this bit to indicate that an interrupt is pending on one of the IN endpoints of the core (in Device mode). The application must read the Device All Endpoints Interrupt (DAINT) register to determine the exact number of the IN endpoint on which the interrupt occurred, and then read the corresponding Device IN Endpoint-n Interrupt (DIEPINTn) register to determine the exact cause of the interrupt. The application must clear the appropriate status bit in the corresponding DIEPINTn register to clear this bit.	Device only	1’b0	RO

© 2007 QuickLogic Corporation

Table 8-29: Core Interrupt Register: GINTSTS (Continued)

Field	Description	Mode	Reset	Access
17	Endpoint Mismatch Interrupt (EPMis) **Note**: This interrupt is valid only in shared FIFO operation. Indicates that an IN token has been received for a non-periodic endpoint, but the data for another endpoint is present in the top of the Non-periodic Transmit FIFO and the IN endpoint mismatch count programmed by the application has expired.	Device only	1'b0	R_SS_WC
16	Reserved		1'b0	
15	End of Periodic Frame Interrupt (EOPF) Indicates that the period specified in the Periodic Frame Interval field of the Device Configuration register (DCFG.PerFrInt) has been reached in the current microframe.	Device only	1'b0	R_SS_WC
14	Isochronous OUT Packet Dropped Interrupt (ISOOutDrop) The core sets this bit when it fails to write an isochronous OUT packet into the RxFIFO because the RxFIFO doesn't have enough space to accommodate a maximum packet size packet for the isochronous OUT endpoint.	Device only	1'b0	R_SS_WC
13	Enumeration Done (EnumDone) The core sets this bit to indicate that speed enumeration is complete. The application must read the Device Status (DSTS) register to obtain the enumerated speed.	Device only	1'b0	R_SS_WC
12	USB Reset (USBRst) The core sets this bit to indicate that a reset is detected on the USB.	Device only	1'b0	R_SS_WC
11	USB Suspend (USBSusp) The core sets this bit to indicate that a suspend was detected on the USB. The core enters the Suspended state when there is no activity on the phy_line_state_i signal for an extended period of time.	Device only	1'b0	R_SS_WC
10	Early Suspend (ErlySusp) The core sets this bit to indicate that an Idle state has been detected on the USB for 3 ms.	Device only	1'b0	R_SS_WC
9	i2c compatible serial bus Interrupt (I2CINT) The core sets this interrupt when I^2C access is completed on the i2c compatible serial bus interface.	Host and Device	1'b0	R_SS_WC
8	ULPI Carkit Interrupt (ULPICKINT) The core sets this interrupt when a ULPI Carkit interrupt is received. The core's PHY sets ULPI Carkit interrupt in UART or Audio mode. I2C Carkit Interrupt (I2CCKINT) The core sets this interrupt when a Carkit interrupt is received. The core's PHY sets the I^2C Carkit interrupt in Audio mode.	Host and Device	1'b0	R_SS_WC
7	Global OUT NAK Effective (GOUTNakEff) Indicates that the Set Global OUT NAK bit in the Device Control register (DCTL.SGOUTNak), set by the application, has taken effect in the core. This bit can be cleared by writing the Clear Global OUT NAK bit in the Device Control register (DCTL.CGOUTNak).	Device only	1'b0	RO
6	Global IN Non-periodic NAK Effective (GINNakEff) Indicates that the Set Global Non-periodic IN NAK bit in the Device Control register (DCTL.SGNPInNak), set by the application, has taken effect in the core. That is, the core has sampled the Global IN NAK bit set by the application. This bit can be cleared by clearing the Clear Global Non-periodic IN NAK bit in the Device Control register (DCTL.CGNPInNak). This interrupt does not necessarily mean that a NAK handshake is sent out on the USB. The STALL bit takes precedence over the NAK bit.	Device only	1'b0	RO

© 2007 QuickLogic Corporation

Table 8-29: Core Interrupt Register: GINTSTS (Continued)

Field	Description	Mode	Reset	Access
5	Non-periodic TxFIFO Empty (NPTxFEmp) This interrupt is valid only when it is = 0. This interrupt is asserted when the Non-periodic TxFIFO is either half or completely empty, and there is space for at least one entry to be written to the Non-periodic Transmit Request Queue. The half or completely empty status is determined by the Non-periodic TxFIFO Empty Level bit in the Core Internal Bus Configuration register (GAHBCFG.NPTxFEmpLvl).	Host and Device	1'b0	RO
4	RxFIFO Non-Empty (RxFLvl) Indicates that there is at least one packet pending to be read from the RxFIFO.	Host and Device	1'b0	RO
3	Start of (micro)Frame (Sof) In Host mode, the core sets this bit to indicate that an SOF (FS), micro-SOF (HS), or Keep-Alive (LS) is transmitted on the USB. The application must write a 1 to this bit to clear the interrupt. In Device mode, in the core sets this bit to indicate that an SOF token has been received on the USB. The application can read the Device Status register to get the current (micro)frame number. This interrupt is seen only when the core is operating at either HS or FS.	Host and Device	1'b0	R_SS_WC
2	OTG Interrupt (OTGInt) The core sets this bit to indicate an OTG protocol event. The application must read the OTG Interrupt Status (GOTGINT) register to determine the exact event that caused this interrupt. The application must clear the appropriate status bit in the GOTGINT register to clear this bit.	Host and Device	1'b0	RO
1	Mode Mismatch Interrupt (ModeMis) The core sets this bit when the application is trying to access: A Host mode register, when the core is operating in Device mode A Device mode register, when the core is operating in Host mode The register access is completed on the Internal Bus with an OKAY response, but is ignored by the core internally and doesn't affect the operation of the core.	Host and Device	1'b0	R_SS_WC
0	Current Mode of Operation (CurMod) Indicates the current mode of operation. 1'b0: Device mode 1'b1: Host mode	Host and Device	1'b0	RO

8.3.3.1.6 Core Interrupt Mask Register (GINTMSK)

Offset: 018h

This register works with the Core Interrupt register to interrupt the application. When an interrupt bit is masked, the interrupt associated with that bit is not generated. However, the Core Interrupt (GINTSTS) register bit corresponding to that interrupt is still set.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

© 2007 QuickLogic Corporation

Table 8-30: Core Interrupt Mask Register: GINTMSK

Field	Description	Mode	Reset	Access
31	Resume/Remote Wakeup Detected Interrupt Mask (WkUpIntMsk)	Host and Device	1'b0	R_W
30	Session Request/New Session Detected Interrupt Mask (SessReqIntMsk)	Host and Device	1'b0	R_W
29	Disconnect Detected Interrupt Mask (DisconnIntMsk)	Host and Device	1'b0	R_W
28	Connector ID Status Change Mask (ConIDStsChngMsk)	Host and Device	1'b0	R_W
27	Reserved		1'b0	
26	Periodic TxFIFO Empty Mask (PTxFEmpMsk)	Host only	1'b0	R_W
25	Host Channels Interrupt Mask (HChIntMsk)	Host only	1'b0	R_W
24	Host Port Interrupt Mask (PrtIntMsk)	Host only	1'b0	R_W
23	Reserved		1'b0	
22	Data Fetch Suspended Mask (FetSuspMsk)	Device only	1'b0	R_W
21	Incomplete Periodic Transfer Mask (incomplPMsk) Incomplete Isochronous OUT Transfer Mask (incompISOOUTMsk)	Host only Device only	1'b0	R_W
20	Incomplete Isochronous IN Transfer Mask (incompISOINMsk)	Device only	1'b0	R_W
19	OUT Endpoints Interrupt Mask (OEPIntMsk)	Device only	1'b0	R_W
18	IN Endpoints Interrupt Mask (INEPIntMsk)	Device only	1'b0	R_W
17	Endpoint Mismatch Interrupt Mask (EPMisMsk)	Device only	1'b0	R_W
16	Reserved		1'b0	
15	End of Periodic Frame Interrupt Mask (EOPFMsk)	Device only	1'b0	R_W
14	Isochronous OUT Packet Dropped Interrupt Mask (ISOOutDropMsk)	Device only	1'b0	R_W
13	Enumeration Done Mask (EnumDoneMsk)	Device only	1'b0	R_W
12	USB Reset Mask (USBRstMsk)	Device only	1'b0	R_W
11	USB Suspend Mask (USBSuspMsk)	Device only	1'b0	R_W
10	Early Suspend Mask (ErlySuspMsk)	Device only	1'b0	R_W
9	I^2C Interrupt Mask (I2CINT)	Host and Device	1'b0	R_W
8	ULPI Carkit Interrupt Mask (ULPICKINTMsk) I^2C Carkit Interrupt Mask (I2CCKINTMsk)	Host and Device	1'b0	R_W
7	Global OUT NAK Effective Mask (GOUTNakEffMsk)	Device only	1'b0	R_W
6	Global Non-periodic IN NAK Effective Mask (GINNakEffMsk)	Device only	1'b0	R_W
5	Non-periodic TxFIFO Empty Mask (NPTxFEmpMsk)	Host and Device	1'b0	R_W
4	Receive FIFO Non-Empty Mask (RxFLvlMsk)	Host and Device	1'b0	R_W
3	Start of (micro)Frame Mask (SofMsk)	Host and Device	1'b0	R_W
2	OTG Interrupt Mask (OTGIntMsk)	Host and Device	1'b0	R_W
1	Mode Mismatch Interrupt Mask (ModeMisMsk)	Host and Device	1'b0	R_W
0	Reserved		1'b0	

© 2007 QuickLogic Corporation

8.3.3.1.7 Receive Status Debug Read/Status Read and Pop Registers (GRXSTSR/GRXSTSP)

Offset for Read: 01Ch

Offset for Pop: 020h

A read to the Receive Status Debug Read register returns the contents of the top of the Receive FIFO. A read to the Receive Status Read and Pop register additionally pops the top data entry out of the RxFIFO.

The receive status contents must be interpreted differently in Host and Device modes. The core ignores the receive status pop/read when the receive FIFO is empty and returns a value of 32'h0000_0000. The application must only pop the Receive Status FIFO when the Receive FIFO Non-Empty bit of the Core Interrupt register (GINTSTS.RxFLvl) is asserted.

Table 8-31 shows the use of these registers in Host mode, and **Table 8-32** shows Device mode.

Table 8-31: Host Mode Receive Status Debug Read/Status Read and Pop Registers: GRXSTSR/GRXSTSP

Field	Description	Reset	Access
31:21	Reserved	11'h0	
20:17	Packet Status (PktSts) Indicates the status of the received packet 4'b0010: IN data packet received 4'b0011: IN transfer completed (triggers an interrupt) 4'b0101: Data toggle error (triggers an interrupt) 4'b0111: Channel halted (triggers an interrupt) Others: Reserved	4'b0	RO
16:15	Data PID (DPID) Indicates the Data PID of the received packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA	2'b0	RO
14:4	Byte Count (BCnt) Indicates the byte count of the received IN data packet.	11'h0	RO
3:0	Channel Number (ChNum) Indicates the channel number to which the current received packet belongs.	4'h0	RO

© 2007 QuickLogic Corporation

Table 8-32: Device Mode Receive Status Debug Read/Status Read and Pop Registers: GRXSTSR/GRXSTSP

Field	Description	Reset	Access
31:25	Reserved	7'h0	
24:21	Frame Number (FN) This is the least significant 4 bits of the (micro)frame number in which the packet is received on the USB. This field is supported only when isochronous OUT endpoints are supported.	4'h0	RO
20:17	Packet Status (PktSts) Indicates the status of the received packet 4'b0001: Global OUT NAK (triggers an interrupt) 4'b0010: OUT data packet received 4'b0011: OUT transfer completed (triggers an interrupt) 4'b0100: SETUP transaction completed (triggers an interrupt) 4'b0110: SETUP data packet received Others: Reserved	4'h0	RO
16:15	Data PID (DPID) Indicates the Data PID of the received OUT data packet 2'b00: DATA0 2'b10: DATA1 2'b01: DATA2 2'b11: MDATA	2'b0	RO
14:4	Byte Count (BCnt) Indicates the byte count of the received data packet.	11'h0	RO
3:0	Endpoint Number (EPNum) Indicates the endpoint number to which the current received packet belongs.	4'h0	RO

8.3.3.1.8 Receive FIFO Size Register (GRXFSIZ)

Offset: 024h

The application can program the RAM size that must be allocated to the RxFIFO.

Table 8-33: Receive FIFO Size Register: GRXFSIZ

Field	Description	Reset	Access
31:16	Reserved	16'h0	
15:0	RxFIFO Depth (RxFDep) This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 544	16'h0220	RO / R_W

8.3.3.2 Non-Periodic Transmit FIFO Size Register (GNPTXFSIZ)

Offset: 028h

The application can program the RAM size and the memory start address for the Non-periodic TxFIFO.

Table 8-34: Non-Periodic Transmit FIFO Size Register: GNPTXFSIZ

Field	Description	Reset	Access
31:16	**Non-periodic TxFIFO Depth (NPTxFDep)** For host mode, this field is always valid. Minimum value is 16 Maximum value is 256	16'h0100	RO / R_W
15:0	**Non-periodic Transmit RAM Start Address (NPTxFStAddr)** This field contains the memory start address for Non-periodic Transmit FIFO RAM. IN Endpoint FIFO0 Transmit RAM Start Address (INEPTxF0StAddr)	16'h0220	RO / R_W

8.3.3.2.1 Non-Periodic Transmit FIFO/Queue Status Register (GNPTXSTS)

Offset 02Ch

In Device mode, this register is valid only in Shared FIFO operation.

This read-only register contains the free space information for the Non-periodic TxFIFO and the Non-periodic Transmit Request Queue.

© 2007 QuickLogic Corporation

Table 8-35: Non-Periodic Transmit FIFO/Queue Status Register: GNPTXSTS

Field	Description	Reset	Access
31	Reserved	1'b0	
30:24	Top of the Non-periodic Transmit Request Queue (NPTxQTop) Entry in the Non-periodic Tx Request Queue that is currently being processed by the MAC. Bits [30:27]: Channel/endpoint number Bits [26:25]: 2'b00: IN/OUT token 2'b01: Zero-length transmit packet (device IN/host OUT) 2'b10: PING/CSPLIT token 2'b11: Channel halt command Bit [24]: Terminate (last entry for selected channel/endpoint)	7'h0	RO
23:16	Non-periodic Transmit Request Queue Space Available (NPTxQSpcAvail) Indicates the amount of free space available in the Non-periodic Transmit Request Queue. This queue holds both IN and OUT requests in Host mode. Device mode has only IN requests. 8'h0: Non-periodic Transmit Request Queue is full 8'h1: 1 location available 8'h2: 2 locations available *n*: *n* locations available ($0 \leq n \leq 8$) Others: Reserved	8'h8	RO
15:0	Non-periodic TxFIFO Space Avail (NPTxFSpcAvail) Indicates the amount of free space available in the Non-periodic TxFIFO. Values are in terms of 32-bit words. 16'h0: Non-periodic TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'h*n*: *n* words available (where $0 \leq n \leq 32,768$) 16'h8000: 32,768 words available Others: Reserved	16'h0100	RO

8.3.3.2.2 I²C Access Register (GI2CCTL)

Offset: 030h

The application can use this register to access OTG devices connected to the OTG core through the i2c compatible serial bus interface.

The i2c compatible serial bus interface on the OTG core can read/write the register space in the attached I²C device. The following table describes the register fields.

Table 8-36: I²C Access Register: GI2CCTL

Field	Description	Reset	Access
31	I²C Busy/Done (BsyDne) The application sets this bit to 1'b1 to start a request on the i2c compatible serial bus interface. When the transfer is complete, the core deasserts this bit to 1'b0. As long as the bit is set, indicating that the i2c compatible serial bus interface is busy, the application cannot start another request on the interface.	1'b0	R_WS_SC
30	Read/Write Indicator (RW) Indicates whether a read or write register transfer must be performed on the interface. Read/write bursting is not supported for registers. 1'b1: Read 1'b0: Write	1'b0	R_W
29	Reserved	1'b0	
28	I²C DatSe0 USB Mode (I2CDatSe0) Selects the FS interface USB mode. 1'b0: VP_VM USB mode 1'b0: DAT_SE0 USB mode	1'b0	R_W
27:26	I²C Device Address (I2CDevAdr) Selects the address of the I²C Slave on the USB 1.1 full-speed serial transceiver that the core uses for OTG signaling. 2'b00: 7'h2C 2'b01: 7'h2D 2'b10: 7'h2E 2'b11: 7'h2F	2'b0	R_W
25	I²C Suspend Control (I2CSuspCtl) Selects how Suspend is connected to a full-speed transceiver in I²C mode. 1'b0: Use the dedicated utmi_suspend_n pin 1'b1: Use an I²C write to program the Suspend bit in the PHY register	1'b0	R_W
24	I²C ACK (Ack) Indicates whether an ACK response was received from the I²C Slave. This bit is valid when BsyDne is reset. 1'b0: NAK 1'b1: ACK	1'b0	RO
23	I²C Enable (I2CEn) Enables the I²C Master to initiate I²C transactions on the i2c compatible serial bus interface.	1'b0	R_W
22:16	I²C Address (Addr) This is the 7-bit I²C device address used by software to access any external I²C Slave, including the I²C Slave on a USB 1.1 OTG full-speed serial transceiver. Software can change this address to access different I²C Slaves.	7'h0	R_W
15:8	I²C Register Addr (RegAddr) This field programs the address of the register to be read from or written to.	8'h00	R_W
7:0	I²C Read/Write Data (RWData) After a register read operation, this field holds the read data for the application. During a write operation, the application can use this register to program the write data to be written to a register. During writes, this field holds the write data.	8'h00	R_W

© 2007 QuickLogic Corporation

8.3.3.2.3 PHY Vendor Control Register (GPVNDCTL)

Offset: 034h

The application can use this register to access PHY registers.

For a UTMI+ PHY, the USB OTG core uses the UTMI+ Vendor Control interface for PHY register access. For a ULPI PHY, the core uses the ULPI interface for PHY register access. The application sets Vendor Control register for PHY register access and times the PHY register access. The application polls the VStatus Done bit in this register for the completion of the PHY register access.

Table 8-37: PHY Vendor Control Register: GPVNDCTL

Field	Description	Reset	Access
31	Disable ULPI Drivers (DisUlpiDrvr) Software sets this bit when it has finished processing the ULPI Carkit Interrupt (GINTSTS.ULPICKINT). When set, the USB OTG core disables drivers for output signals and masks input signal for the ULPI interface. USB OTG clears this bit before enabling the ULPI interface.	1'b0	R_WS_SC
30:28	Reserved	3'h0	
27	VStatus Done (VStsDone) The core sets this bit when the vendor control access is done. This bit is cleared by the core when the application sets the New Register Request bit (bit 25).	1'b0	R_SS_WC_SC
26	VStatus Busy (VStsBsy) The core sets this bit when the vendor control access is in progress and clears this bit when done.	1'b0	RO
25	New Register Request (NewRegReq) The application sets this bit for a new vendor control access.	1'b0	R_WS_SC
24:23	Reserved	2'h0	
22	Register Write (RegWr) Set this bit for register writes, and clear it for register reads.	1'b0	R_W
21:16	Register Address (RegAddr) The 6-bit PHY register address for immediate PHY Register Set access. Set to 6'h2F for Extended PHY Register Set access.	6'h0	R_W
15:8	UTMI+ Vendor Control Register Address (VCtrl) The 4-bit register address a vendor defined 4-bit parallel output bus. Bits 11:8 of this field are placed on utmi_vcontrol[3:0]. ULPI Extended Register Address (ExtRegAddr) The 6-bit PHY extended register address.	8'h0	R_W
7:0	Register Data (RegData) Contains the write data for register write. Read data for register read, valid when VStatus Done is set.	8'h0	R_W

8.3.3.2.4 General Purpose Input/Output Register (GGPIO)

Offset: 038h

The application can use this register for general purpose input/output ports or for debugging.

Table 8-38: General Purpose Input/Output Register: GGPIO

Field	Description	Reset	Access
31:16	General Purpose Output (GPO) This field is driven as an output from the core, gp_o[15:0]. The application can program this field to determine the corresponding value on the gp_o[15:0] output.	16'h0	R_W
15:0	General Purpose Input (GPI) This field's read value reflects the gp_i[15:0] core input value.	16'h0	RO

8.3.3.2.5 User ID Register (GUID)

Offset: 03Ch

This is a read/write register containing the User ID. This register can be used in the following ways:

- To store the version or revision of your system
- To store hardware configurations that are outside the USB OTG core
- As a scratch register

Table 8-39: User ID Register: GUID

Field	Description	Reset	Access
31:0	User ID (UserID) Application-programmable ID field.	16'h12345678	R_W

8.3.3.2.6 User HW Config1 Register (GHWCFG1)

Offset: 044

This register contains the logical endpoint direction(s) selected.

Table 8-40: User HW Config1 Register: GHWCFG1

Field	Description	Reset	Access
31:0	Endpoint Direction (epdir) Two bits per endpoint represent the direction. 2'b00: BIDIR (IN and OUT) endpoint 2'b01: IN endpoint 2'b10: OUT endpoint 2'b11: Reserved Bits [31:30]: Endpoint 15 direction Bits [29:28]: Endpoint 14 direction ... Bits [3:2]: Endpoint 1 direction Bits[1:0]: Endpoint 0 direction (always BIDIR)	32'h5556AAA0	RO

8.3.3.2.7 User HW Config2 Register (GHWCFG2)

Offset: 048h

© 2007 QuickLogic Corporation

This register contains configuration options selected.

Table 8-41: User HW Config2 Register: GHWCFG2

Field	Description	Reset	Access
31:30	Reserved	2'b0	
29:26	Device Mode IN Token Sequence Learning Queue Depth (TknQDepth)	4'b1110	RO
25:24	Host Mode Periodic Request Queue Depth (PTxQDepth) 2'b10: 8	2'b10	RO
23:22	Non-periodic Request Queue Depth (NPTxQDepth) 2'b10: 8	2'b1	RO
21:20	Reserved	2'b0	
19	Dynamic FIFO Sizing Enabled (DynFifoSizing) 1'b1: Yes	1'b1	RO
18	Periodic OUT Channels Supported in Host Mode (PerioSupport) 1'b1: Yes	1'b1	RO
17:14	Number of Host Channels (NumHstChnl) Indicates the number of host channels supported by the core in Host mode. 15 specifies 16 channels.	4'b1111	RO
13:10	Number of Device Endpoints (NumDevEps) Indicates the number of device endpoints supported by the core in Device mode in addition to control endpoint 0. The range of this field is 1–15.	4'b1111	RO
9:8	Full-Speed PHY Interface Type (FSPhyType) 2'b11: FS pins shared with ULPI pins	2'b11	RO
7:6	High-Speed PHY Interface Type (HSPhyType) 2'b11: UTMI+ and ULPI	2'b11	RO
5	Point-to-Point (SingPnt) 1'b0: Multi-point application	1'b0	RO
4:3	Architecture (OtgArch) 2'b10: Internal DMA	2'b10	RO
2:0	Mode of Operation (OtgMode) 3'b000: HNP- and SRP-Capable OTG (Host & Device)	3'b000	RO

8.3.3.2.8 User HW Config3 Register (GHWCFG3)

Offset: 04Ch

This register contains the configuration options selected.

Table 8-42: User HW Config3 Register: GHWCFG3

Field	Description	Reset	Access
31:16	DFIFO Depth (DfifoDepth) This value is in terms of 32-bit words. Minimum value is 32 Maximum value is 32,768	32'h520	RO
15:13	Reserved	3'b0	
12	Internal Bus and PHY Synchronous (AhbPhySync) Indicates whether Internal Bus and PHY clocks are synchronous to each other. 1'b1: Yes This bit is tied to 1.	1'b1	RO
11	Reset Style for Clocked always Blocks in RTL (RstType) 1'b0: Asynchronous reset is used in the core	1'b0	RO
10	Optional Features Removed (OptFeature) Indicates whether the User ID register, GPIO interface ports, and SOF toggle and counter ports were removed for gate count optimization by enabling Remove Optional Features? 1'b0: No	1'b0	RO
9	Vendor Control Interface Support 1'b1: Vendor Control Interface is available.	1'b1	RO
8	I^2C Selection 1'b1: i2c compatible serial bus Interface is available on the core.	1'b1	RO
7	OTG Function Enabled (OtgEn) The application uses this bit to indicate the USB OTG core's OTG capabilities. 1'b1: OTG Capable	1'b1	RO
6:4	Width of Packet Size Counters (PktSizeWidth) 3'b110: 10 bits	3'b110	RO
3:0	Width of Transfer Size Counters (XferSizeWidth) 4'b1000: 19 bits	4'b1000	RO

8.3.3.2.9 User HW Config4 Register (GHWCFG4)

Offset: 050h

This register contains the configuration options selected.

© 2007 QuickLogic Corporation

Table 8-43: User HW Config4 Register: GHWCFG4

Field	Description	Reset	Access
31:30	Reserved	2'h0	
29:26	Number of Device Mode IN Endpoints Including Control Endpoints (INEps) Range 0 -15 0 : 1 IN Endpoint 1 : 2 IN Endpoints .... 15 : 16 IN Endpoints **NOTE:** A maximum of 9 Enpoints are currently supported	4'h09	RO
25	Enable Dedicated Transmit FIFO for device IN Endpoints (DedFifoMode) Dedicated Transmit FIFO Operation not enabled.	1'b0	RO
24	"session_end" Filter Enabled (SessEndFltr) 1'b0: No filter	1'b0	RO
23	"b_valid" Filter Enabled (BValidFltr) 1'b0: No filter	1'b0	RO
22	"a_valid" Filter Enabled (AValidFltr) 1'b0: No filter	1'b0	RO
21	"vbus_valid" Filter Enabled (VBusValidFltr) 1'b0: No filter	1'b0	RO
20	"iddig" Filter Enable (IddgFltr) 1'b0: No filter	1'b0	RO
19:16	Number of Device Mode Control Endpoints in Addition to Endpoint 0 (NumCtlEps)	1'b0	RO
15:14	UTMI+ PHY/ULPI-to-Internal UTMI+ Wrapper Data Width (PhyDataWidth) When a ULPI PHY is used, an internal wrapper converts ULPI to UTMI+ . 2'b00: 8 bits	2'b00	RO
13:6	Reserved	8'h0	
5	Minimum Internal Bus Frequency Less Than 60 MHz (AhbFreq) 1'b1: Yes	1'b1	RO
4	Enable Power Optimization? (EnablePwrOpt) 1'b1: Yes	1'b1	RO
3:0	Number of Device Mode Periodic IN Endpoints (NumDevPerioEps) Range: 0–15	4'b0010	RO

8.3.3.2.10 Host Periodic Transmit FIFO Size Register (HPTXFSIZ)

Offset: 100h

This register holds the size and the memory start address of the Periodic TxFIFO.

© 2007 QuickLogic Corporation

Table 8-44: Host Periodic Transmit FIFO Size Register: HPTXFSIZ

Field	Description	Reset	Access
31:16	Host Periodic TxFIFO Depth (PTxFSize) This value is in terms of 32-bit words. Minimum value is 16 Maximum value is 512 The power-on reset value of this register is specified as the Largest Host Mode Periodic Tx Data FIFO Depth (parameter OTG_TX_HPERIO_DFIFO_DEPTH) If Enable Dynamic FIFO Sizing? was deselected, these flops are optimized, and reads return the power-on value. If Enable Dynamic FIFO Sizing? was selected, you can write a new value in this field. Programmed values must not exceed the power-on value set.	16'h200	RO / R_W
15:0	Host Periodic TxFIFO Start Address (PTxFStAddr) The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth and Largest Non-periodic Tx Data FIFO Depth specified. These parameters are: OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH If you are programming new values for the RxFIFO or Non-periodic TxFIFO, you can write their sum in this field. Programmed values must not exceed the power-on value set.	16'h0520	RO / R_W

8.3.3.2.11 Device Periodic Transmit FIFO-n Size Register (DPTXFSIZn)

FIFO_number: $1 \leq n \leq 15$

Offset: 104h + (FIFO_number – 1) * 04h

This register is valid only in shared FIFO operation.

This register holds the memory start address of each periodic TxFIFO to be implemented in Device mode. Each periodic FIFO holds the data for one periodic IN endpoint. This register is repeated for each periodic FIFO instantiated.

© 2007 QuickLogic Corporation

Table 8-45: Device Periodic Transmit FIFO-n Register: DPTXFSIZn

Field	Description	Reset	Access
31:16	Device Periodic TxFIFO Size (DPTxFSize) This value is in terms of 32-bit words. Minimum value is 4 Maximum value is 512 The value of this register is the Largest Device Mode Periodic Tx Data FIFO Depth (parameter OTG_TX_DPERIO_DFIFO_DEPTH_n), as specified during configuration.	16'h200	RO
15:0	Device Periodic TxFIFO RAM Start Address (DPTxFStAddr) Holds the start address in the RAM for this periodic FIFO. The power-on reset value of this register is the sum of the Largest Rx Data FIFO Depth, Largest Non-periodic Tx Data FIFO Depth, and all lower numbered Largest Device Mode Periodic Tx Data FIFOn Depth specified during configuration. These parameters are: OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH + SUM 1 to *n* – 1 (OTG_TX_DPERIO_DFIFO_DEPTH_n). When *n* = 1, the expression above becomes OTG_RX_DFIFO_DEPTH + OTG_TX_NPERIO_DFIFO_DEPTH. If you are programming new values for the RxFIFO Non-periodic TxFIFO, or device Periodic TxFIFOs, you can write their sum in this field. Programmed values must not exceed the power-on value.	16'h0320	RO / R_W

8.3.3.3 Host Mode Registers

These registers affect the operation of the core in the Host mode. Host mode registers must not be accessed in Device mode, as the results are undefined. Host Mode registers can be categorized as follows:

- Host Global Registers
  - 'Host Configuration Register (HCFG)" on page 114
  - 'Host Frame Interval Register (HFIR)" on page 114
  - 'Host Frame Number/Frame Time Remaining Register (HFNUM)" on page 115
  - 'Host Periodic Transmit FIFO/Queue Status Register (HPTXSTS)" on page 116
  - 'Host All Channels Interrupt Register (HAINT)" on page 117
  - 'Host All Channels Interrupt Mask Register (HAINTMSK)" on page 117
  - 'Host Port Control and Status Register (HPRT)" on page 117
- Host Port Control and Status Register
  - 'Host Channel-n Characteristics Register (HCCHARn)" on page 120
- Host Channel-Specific Registers
  - 'Host Channel-n Characteristics Register (HCCHARn)" on page 120
  - 'Host Channel-n Split Control Register (HCSPLTn)" on page 121
  - 'Host Channel-n Interrupt Register (HCINTn)" on page 121

© 2007 QuickLogic Corporation

– 'Host Channel-n Interrupt Mask Register (HCINTMSKn)" on page 122
– 'Host Channel-n Transfer Size Register (HCTSIZn)" on page 123
– 'Host Channel-n DMA Address Register (HCDMAn)" on page 123

Host Global Registers

8.3.3.3.1 Host Configuration Register (HCFG)

Offset: 400h

This register configures the core after power-on. Do not make changes to this register after initializing the host.

Table 8-46: Host Configuration Register: HCFG

Field	Description	Reset	Access
31:3	Reserved	30'h0	
2	FS- and LS-Only Support (FSLSSupp) The application uses this bit to control the core's enumeration speed. Using this bit, the application can make the core enumerate as a FS host, even if the connected device supports HS traffic. Do not make changes to this field after initial programming. 1'b0: HS/FS/LS, based on the maximum speed supported by the connected device 1'b1: FS/LS-only, even if the connected device can support HS	1'b0	R_W
1:0	FS/LS PHY Clock Select (FSLSPclkSel) When the core is in FS Host mode 2'b00: PHY clock is running at 30/60 MHz 2'b01: PHY clock is running at 48 MHz Others: Reserved When the core is in LS Host mode 2'b00: PHY clock is running at 30/60 MHz. When the UTMI+ /ULPI PHY Low Power mode is not selected, use 30/60 MHz. 2'b01: PHY clock is running at 48 MHz. When the UTMI+ PHY Low Power mode is selected, use 48MHz if the PHY supplies a 48 MHz clock during LS mode. 2'b10: PHY clock is running at 6 MHz. In USB 1.1 FS mode, use 6 MHz when the UTMI+ PHY Low Power mode is selected and the PHY supplies a 6 MHz clock during LS mode. If you select a 6 MHz clock during LS mode, you must do a soft reset. 2'b11: Reserved	2'b0	R_W

8.3.3.3.2 Host Frame Interval Register (HFIR)

Offset: 404h

This register stores the frame interval information for the current speed to which the USB OTG core has enumerated.

© 2007 QuickLogic Corporation

Table 8-47: Host Frame Interval Register: HFIR

Field	Description	Reset	Access
31:16	Reserved	16'h0	
15:0	Frame Interval (FrInt) The value that the application programs to this field specifies the interval between two consecutive SOFs (FS) or micro-SOFs (HS) or Keep-Alive tokens (HS). This field contains the number of PHY clocks that constitute the required frame interval. The default value set in this field for a FS operation when the PHY clock frequency is 60 MHz. The application can write a value to this register only after the Port Enable bit of the Host Port Control and Status register (HPRT.PrtEnaPort) has been set. If no value is programmed, the core calculates the value based on the PHY clock specified in the FS/LS PHY Clock Select field of the Host Configuration register (HCFG.FSLSPclkSel). Do not change the value of this field after the initial configuration. 125 µs * (PHY clock frequency for HS) 1 ms * (PHY clock frequency for FS/LS)	16'd60000	R_W

8.3.3.3.3 Host Frame Number/Frame Time Remaining Register (HFNUM)

Offset: 408h

This register indicates the current frame number. It also indicates the time remaining (in terms of the number of PHY clocks) in the current (micro) frame.

Table 8-48: Host Frame Number/Frame Time Remaining Register: HFNUM

Field	Description	Reset	Access
31:16	Frame Time Remaining (FrRem) Indicates the amount of time remaining in the current microframe (HS) or frame (FS/LS), in terms of PHY clocks. This field decrements on each PHY clock. When it reaches zero, this field is reloaded with the value in the Frame Interval register and a new SOF is transmitted on the USB.	16'h0	RW
15:0	Frame Number (FrNum) This field increments when a new SOF is transmitted on the USB, and is reset to 0 when it reaches 16'h3FFF. This field is writable only if Remove Optional Features? was not selected. Otherwise, reads return the frame number value.	16'h3FFF	RO

© 2007 QuickLogic Corporation

8.3.3.3.4 Host Periodic Transmit FIFO/Queue Status Register (HPTXSTS)

Offset: 410h

This read-only register contains the free space information for the Periodic TxFIFO and the Periodic Transmit Request Queue.

Table 8-49: Host Periodic Transmit FIFO/Queue Status Register: HPTXSTS

Field	Description	Reset	Access
31:24	Top of the Periodic Transmit Request Queue (PTxQTop) This indicates the entry in the Periodic Tx Request Queue that is currently being processes by the MAC. This register is used for debugging. Bit [31]: Odd/Even (micro)frame 1'b0: send in even (micro)frame 1'b1: send in odd (micro)frame Bits [30:27]: Channel/endpoint number Bits [26:25]: Type 2'b00: IN/OUT 2'b01: Zero-length packet 2'b10: CSPLIT 2'b11: Disable channel command Bit [24]: Terminate (last entry for the selected channel/endpoint)	8'h0	RO
23:16	Periodic Transmit Request Queue Space Available (PTxQSpcAvail) Indicates the number of free locations available to be written in the Periodic Transmit Request Queue. This queue holds both IN and OUT requests. 8'h0: Periodic Transmit Request Queue is full 8'h1: 1 location available 8'h2: 2 locations available *n*: *n* locations available ($0 \leq n \leq 8$) Others: Reserved	8'h8	RO
15:0	Periodic Transmit Data FIFO Space Available (PTxFSpcAvail) Indicates the number of free locations available to be written to in the Periodic TxFIFO. Values are in terms of 32-bit words 16'h0: Periodic TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'h*n*: *n* words available (where $0 \leq n \leq 32{,}768$) 16'h8000: 32,768 words available Others: Reserved	16'h0100	RW

© 2007 QuickLogic Corporation

8.3.3.3.5 Host All Channels Interrupt Register (HAINT)

Offset: 414h

When a significant event occurs on a channel, the Host All Channels Interrupt register interrupts the application using the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt). This is shown in Figure 8-3. There is one interrupt bit per channel, up to a maximum of 16 bits. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Host Channel-n Interrupt register.

Table 8-50: Host All Channels Interrupt Register: HAINT

Field	Description	Reset	Access
31:16	Reserved	16'h0	
15:0	Channel Interrupts (HAINT) One bit per channel: Bit 0 for Channel 0, bit 15 for Channel 15	16'h0	RO

8.3.3.3.6 Host All Channels Interrupt Mask Register (HAINTMSK)

Offset: 418h

The Host All Channel Interrupt Mask register works with the Host All Channel Interrupt register to interrupt the application when an event occurs on a channel. There is one interrupt mask bit per channel, up to a maximum of 16 bits.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Table 8-51: Host All Channels Interrupt Mask Register: HAINTMSK

Field	Description	Reset	Access
31:16	Reserved	16'h0	
15:0	Channel Interrupt Mask (HAINTMsk) One bit per channel: Bit 0 for channel 0, bit 15 for channel 15	16'h0	R_W

Host Port Control and Status Register

8.3.3.3.7 Host Port Control and Status Register (HPRT)

Offset: 440h

This register is available only in Host mode. Currently, the OTG Host supports only one port.

A single register holds USB port-related information such as USB reset, enable, suspend, resume, connect status, and test mode for each port. It is shown in Figure 8-3. The R_SS_WC bits in this register can trigger an interrupt to the application through the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). On a Port Interrupt, the application must read this register and clear the bit that caused the interrupt. For the R_SS_WC bits, the application must write a 1 to the bit to clear the interrupt.

© 2007 QuickLogic Corporation

Table 8-52: Host Port Control and Status Register: HPRT

Field	Description	Reset	Access
31:19	Reserved	12'h0	
18:17	Port Speed (PrtSpd) Indicates the speed of the device attached to this port. 2'b00: High speed 2'b01: Full speed 2'b10: Low speed 2'b11: Reserved	2'b0	RO
16:13	Port Test Control (PrtTstCtl) The application writes a nonzero value to this field to put the port into a Test mode, and the corresponding pattern is signaled on the port. 4'b0000: Test mode disabled 4'b0001: Test_J mode 4'b0010: Test_K mode 4'b0011: Test_SE0_NAK mode 4'b0100: Test_Packet mode 4'b0101: Test_Force_Enable Others: Reserved	4'h0	R_W
12	Port Power (PrtPwr) The application uses this field to control power to this port, and the core clears this bit on an overcurrent condition. 1'b0: Power off 1'b1: Power on	1'b0	R_W_SC
11:10	Port Line Status (PrtLnSts) Indicates the current logic level USB data lines Bit [10]: Logic level of D– Bit [11]: Logic level of D +	2'b0	RO
9	Reserved	1'b0	
8	Port Reset (PrtRst) When the application sets this bit, a reset sequence is started on this port. The application must time the reset period and clear this bit after the reset sequence is complete. 1'b0: Port not in reset 1'b1: Port in reset The application must leave this bit set for at least a minimum duration mentioned below to start a reset on the port. The application can leave it set for another 10 ms in addition to the required minimum duration, before clearing the bit, even though there is no maximum limit set by the USB standard. High speed: 50 ms Full speed/Low speed: 10 ms	1'b0	R_W

© 2007 QuickLogic Corporation

Table 8-52: Host Port Control and Status Register: HPRT (Continued)

Field	Description	Reset	Access
7	Port Suspend (PrtSusp) The application sets this bit to put this port in Suspend mode. The core only stops sending SOFs when this is set. To stop the PHY clock, the application must set the Port Clock Stop bit, which asserts the suspend input pin of the PHY. The read value of this bit reflects the current suspend status of the port. This bit is cleared by the core after a remote wakeup signal is detected or the application sets the Port Reset bit or Port Resume bit in this register or the Resume/Remote Wakeup Detected Interrupt bit or Disconnect Detected Interrupt bit in the Core Interrupt register (GINTSTS.WkUpInt or GINTSTS.DisconnInt, respectively). 1'b0: Port not in Suspend mode 1'b1: Port in Suspend mode	1'b0	R_WS_SC
6	Port Resume (PrtRes) The application sets this bit to drive resume signaling on the port. The core continues to drive the resume signal until the application clears this bit. If the core detects a USB remote wakeup sequence, as indicated by the Port Resume/Remote Wakeup Detected Interrupt bit of the Core Interrupt register (GINTSTS.WkUpInt), the core starts driving resume signaling without application intervention and clears this bit when it detects a disconnect condition. The read value of this bit indicates whether the core is currently driving resume signaling. 1'b0: No resume driven 1'b1: Resume driven	1'b0	R_W_SS_SC
5	Port Overcurrent Change (PrtOvrCurrChng) The core sets this bit when the status of the Port Overcurrent Active bit (bit 4) in this register changes.	1'b0	R_SS_WC
4	Port Overcurrent Active (PrtOvrCurrAct) Indicates the overcurrent condition of the port. 1'b0: No overcurrent condition 1'b1: Overcurrent condition	1'b0	RO
3	Port Enable/Disable Change (PrtEnChng) The core sets this bit when the status of the Port Enable bit [2] of this register changes.	1'b0	R_SS_WC
2	Port Enable (PrtEna) A port is enabled only by the core after a reset sequence, and is disabled by an overcurrent condition, a disconnect condition, or by the application clearing this bit. The application cannot set this bit by a register write. It can only clear it to disable the port. This bit does not trigger any interrupt to the application. 1'b0: Port disabled 1'b1: Port enabled	1'b0	R_SS_SC_WC
1	Port Connect Detected (PrtConnDet) The core sets this bit when a device connection is detected to trigger an interrupt to the application using the Host Port Interrupt bit of the Core Interrupt register (GINTSTS.PrtInt). The application must write a 1 to this bit to clear the interrupt.	1'b0	R_SS_WC
0	Port Connect Status (PrtConnSts) 0: No device is attached to the port. 1: A device is attached to the port.	1'b0	RO

© 2007 QuickLogic Corporation

Host Channel-Specific Registers

8.3.3.3.8 Host Channel-n Characteristics Register (HCCHARn)

Channel_number: $0 \leq n \leq 15$

Offset: 500h + (Channel_number * 20h)

Table 8-53: Host Channel-n Characteristics Register: HCCHARn

Field	Description	Reset	Access
31	Channel Enable (ChEna) This field is set by the application and cleared by the OTG host. 1'b0: Channel disabled 1'b1: Channel enabled	1'b0	R_WS_SC
30	Channel Disable (ChDis) The application sets this bit to stop transmitting/receiving data on a channel, even before the transfer for that channel is complete. The application must wait for the Channel Disabled interrupt before treating the channel as disabled.	1'b0	R_WS_SC
29	Odd Frame (OddFrm) This field is set (reset) by the application to indicate that the OTG host must perform a transfer in an odd (micro)frame. This field is applicable for only periodic (isochronous and interrupt) transactions. 1'b0: Even (micro)frame 1'b1: Odd (micro)frame	1'b0	R_W
28:22	Device Address (DevAddr) This field selects the specific device serving as the data source or sink.	7'h0	R_W
21:20	Multi Count (MC) / Error Count (EC) When the Split Enable bit of the Host Channel-n Split Control register (HCSPLTn.SpltEna) is reset (1'b0), this field indicates to the host the number of transactions that must be executed per microframe for this endpoint. 2'b00: Reserved This field yields undefined results. 2'b01: 1 transaction 2'b10: 2 transactions to be issued for this endpoint per microframe 2'b11: 3 transactions to be issued for this endpoint per microframe When HCSPLTn.SpltEna is set (1'b1), this field indicates the number of immediate retries to be performed for a periodic split transactions on transaction errors. This field must be set to at least 2'b01.	2'b0	R_W
19:18	Endpoint Type (EPType) Indicates the transfer type selected. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt	2'b0	R_W
17	Low-Speed Device (LSpdDev) This field is set by the application to indicate that this channel is communicating to a low-speed device.	1'b0	R_W
16	Reserved	1'b0	

© 2007 QuickLogic Corporation

Table 8-53: Host Channel-n Characteristics Register: HCCHARn (Continued)

Field	Description	Reset	Access
15	Endpoint Direction (EPDir) Indicates whether the transaction is IN or OUT. 1'b0: OUT 1'b1: IN	1'b0	R_W
14:11	Endpoint Number (EPNum) Indicates the endpoint number on the device serving as the data source or sink.	4'h0	R_W
10:0	Maximum Packet Size (MPS) Indicates the maximum packet size of the associated endpoint.	11'h0	R_W

8.3.3.3.9 Host Channel-n Split Control Register (HCSPLTn)

Channel_number: $0 \leq n \leq 15$

Offset: 504h + (Channel_number * 20h)

Table 8-54: Host Channel-n Split Control Register: HCSPLTn

Field	Description	Reset	Access
31	Split Enable (SpltEna) The application sets this field to indicate that this channel is enabled to perform split transactions.	1'b0	R_W
30:17	Reserved	14'h0	
16	Do Complete Split (CompSplt) The application sets this field to request the OTG host to perform a complete split transaction.	1'b0	R_W
15:14	Transaction Position (XactPos) This field is used to determine whether to send all, first, middle, or last payloads with each OUT transaction. 2'b11: All. This is the entire data payload is of this transaction (which is less than or equal to 188 bytes). 2'b10: Begin. This is the first data payload of this transaction (which is larger than 188 bytes). 2'b00: Mid. This is the middle payload of this transaction (which is larger than 188 bytes). 2'b01: End. This is the last payload of this transaction (which is larger than 188 bytes).	2'h0	R_W
13:7	Hub Address (HubAddr) This field holds the device address of the transaction translator's hub.	7'h0	R_W
6:0	Port Address (PrtAddr) This field is the port number of the recipient transaction translator.	7'h0	R_W

8.3.3.3.10 Host Channel-n Interrupt Register (HCINTn)

Channel_number: $0 \leq n \leq 15$

Offset: 508h + (Channel_number * 20h)

This register indicates the status of a channel with respect to USB- and Internal Bus-related events. It is shown in Figure 8-3. The application must read this register when the Host Channels Interrupt bit of the Core Interrupt register (GINTSTS.HChInt) is set. Before the application can read this register, it must first read the Host All Channels Interrupt (HAINT)

register to get the exact channel number for the Host Channel-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the HAINT and GINTSTS registers.

Table 8-55: Host Channel-n Interrupt Register: HCINTn

Field	Description	Reset	Access
31:11	Reserved	21'h0	
10	Data Toggle Error (DataTglErr)	1'b0	R_SS_WC
9	Frame Overrun (FrmOvrun)	1'b0	R_SS_WC
8	Babble Error (BblErr)	1'b0	R_SS_WC
7	Transaction Error (XactErr) Indicates one of the following errors occurred on the USB. CRC check failure Timeout Bit stuff error False EOP	1'b0	R_SS_WC
6	NYET Response Received Interrupt (NYET)	1'b0	R_SS_WC
5	ACK Response Received Interrupt (ACK)	1'b0	R_SS_WC
4	NAK Response Received Interrupt (NAK)	1'b0	R_SS_WC
3	STALL Response Received Interrupt (STALL)	1'b0	R_SS_WC
2	Internal Bus Error (AHBErr) This is generated only in Internal DMA mode when there is an Internal Bus error during Internal Bus read/write. The application can read the corresponding channel's DMA address register to get the error address.	1'b0	R_SS_WC
1	Channel Halted (ChHltd) Indicates the transfer completed abnormally either because of any USB transaction error or in response to disable request by the application.	1'b0	R_SS_WC
0	Transfer Completed (XferCompl) Transfer completed normally without any errors.	1'b0	R_SS_WC

8.3.3.3.11 Host Channel-n Interrupt Mask Register (HCINTMSKn)

Channel_number: $0 \leq n \leq 15$

Offset: 50Ch + (Channel_number * 20h)

This register reflects the mask for each channel status described in the previous section.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Table 8-56: Host Channel-n Interrupt Mask Register: HCINTMSKn

Field	Description	Reset	Access
31:11	Reserved	21'h0	
10	Data Toggle Error Mask (DataTglErrMsk)	1'b0	R_W
9	Frame Overrun Mask (FrmOvrunMsk)	1'b0	R_W
8	Babble Error Mask (BblErrMsk)	1'b0	R_W
7	Transaction Error Mask (XactErrMsk)	1'b0	R_W
6	NYET Response Received Interrupt Mask (NyetMsk)	1'b0	R_W

© 2007 QuickLogic Corporation

Table 8-56: Host Channel-n Interrupt Mask Register: HCINTMSKn (Continued)

Field	Description	Reset	Access
5	ACK Response Received Interrupt Mask (AckMsk)	1'b0	R_W
4	NAK Response Received Interrupt Mask (NakMsk)	1'b0	R_W
3	STALL Response Received Interrupt Mask (StallMsk)	1'b0	R_W
2	Internal Bus Error Mask (AHBErrMsk)	1'b0	R_W
1	Channel Halted Mask (ChHltdMsk)	1'b0	R_W
0	Transfer Completed Mask (XferComplMsk)	1'b0	R_W

8.3.3.3.12 Host Channel-n Transfer Size Register (HCTSIZn)

Channel_number: $0 \leq n \leq 15$

Offset: 510h + (Channel_number * 20h)

Table 8-57: Host Channel-n Transfer Size Register: HCTSIZn

Field	Description	Reset	Access
31	Do Ping (DoPng) Setting this field to 1 directs the host to do PING protocol.	1'h0	R_W
30:29	PID (Pid) The application programs this field with the type of PID to use for the initial transaction. The host maintains this field for the rest of the transfer. 2'b00: DATA0 2'b01: DATA2 2'b10: DATA1 2'b11: MDATA (non-control)/SETUP (control)	2'b0	R_W
28:19	Packet Count (PktCnt) This field is programmed by the application with the expected number of packets to be transmitted (OUT) or received (IN). The host decrements this count on every successful transmission or reception of an OUT/IN packet. Once this count reaches zero, the application is interrupted to indicate normal completion. The width of this counter is specified as Width of Packet Counters.	10'b0	R_W
18:0	Transfer Size (XferSize) For an OUT, this field is the number of data bytes the host sends during the transfer. For an IN, this field is the buffer size that the application has Reserved for the transfer. The application is expected to program this field as an integer multiple of the maximum packet size for IN transactions (periodic and non-periodic). The width of this counter is specified as Width of Transfer Size Counters during configuration.	19'b0	R_W

8.3.3.3.13 Host Channel-n DMA Address Register (HCDMAn)

Channel_number: $0 \leq n \leq 15$

Offset: 514h + (Channel_number * 20h)

This register is used by the OTG host in the internal DMA mode to maintain the current buffer pointer for IN/OUT transactions. The starting DMA address must be DWORD-aligned.

© 2007 QuickLogic Corporation

Table 8-58: Host Channel-n DMA Address Register: HCDMAn

Field	Description	Reset	Access
31:0	DMA Address (DMAAddr) This field holds the start address in the external memory from which the data for the endpoint must be fetched or to which it must be stored. This register is incremented on every Internal Bus transaction.	32'h0	R_W

8.3.3.4 Device Mode Registers

These registers are visible only in Device mode and must not be accessed in Host mode, as the results are unknown. Some of them affect all the endpoints uniformly, while others affect only a specific endpoint. Device Mode registers fall into two categories:

Device Global Registers

- 'Device Configuration Register (DCFG)" on page 125
- 'Device Control Register (DCTL)" on page 126
- 'Device Status Register (DSTS)" on page 128
- 'Device IN Endpoint Common Interrupt Mask Register (DIEPMSK)" on page 129
- 'Device OUT Endpoint Common Interrupt Mask Register (DOEPMSK)" on page 129
- 'Device All Endpoints Interrupt Register (DAINT)" on page 130
- 'Device All Endpoints Interrupt Mask Register (DAINTMSK)" on page 130
- 'Device IN Token Sequence Learning Queue Read Register 1 (DTKNQR1)" on page 131
- 'Device IN Token Sequence Learning Queue Read Register 2 (DTKNQR2)" on page 131
- 'Device IN Token Sequence Learning Queue Read Register 3 (DTKNQR3)" on page 131
- 'Device IN Token Sequence Learning Queue Read Register 4 (DTKNQR4)" on page 132
- 'Device VBUS Discharge Time Register (DVBUSDIS)" on page 132
- 'Device VBUS Pulsing Time Register (DVBUSPULSE)" on page 132
- 'Device IN Endpoint FIFO Empty Interrupt Mask Register: (DIEPEMPMSK)" on page 133
- 'Device Control IN Endpoint 0 Control Register (DIEPCTL0)" on page 133

Device Logical Endpoint-Specific Registers

- 'Device Control OUT Endpoint 0 Control Register (DOEPCTL0)" on page 134
- 'Device Endpoint-n Control Register (DIEPCTLn/DOEPCTLn)" on page 136
- 'Device Endpoint-n Interrupt Register (DIEPINTn/DOEPINTn)" on page 140
- 'Device Endpoint 0 Transfer Size Register (DIEPTSIZ0/DOEPTSIZ0)" on page 141
- 'Device Endpoint-n Transfer Size Register (DIEPTSIZn/DOEPTSIZn)" on page 143

© 2007 QuickLogic Corporation

- 'Device Endpoint-n DMA Address Register (DIEPDMAn/DOEPDMAn)" on page 144
- 'Device IN Endpoint Transmit FIFO Status Register (DTXFSTSn)" on page 145

Device Global Registers

8.3.3.4.1 Device Configuration Register (DCFG)

Offset: 800h

This register configures the core in Device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

Table 8-59: Device Configuration Register: DCFG

Field	Description	Reset	Access
31:23	Reserved	9'h0	
22:18	IN Endpoint Mismatch Count (EPMisCnt) This field is valid only in shared FIFO operation. The application programs this filed with a count that determines when the core generates an Endpoint Mismatch interrupt (GINTSTS.EPMis). The core loads this value into an internal counter and decrements it. The counter is reloaded whenever there is a match or when the counter expires. The width of this counter depends on the depth of the Token Queue.	5'h8	R_W
17:13	Reserved	5'h0	
12:11	Periodic Frame Interval (PerFrInt) Indicates the time within a (micro)frame at which the application must be notified using the End Of Periodic Frame Interrupt. This can be used to determine if all the isochronous traffic for that (micro)frame is complete. 2'b00: 80% of the (micro)frame interval 2'b01: 85% 2'b10: 90% 2'b11: 95%	2'h0	R_W
10:4	Device Address (DevAddr) The application must program this field after every SetAddress control command.	7'h0	R_W

Table 8-59: Device Configuration Register: DCFG (Continued)

Field	Description	Reset	Access
3	Reserved	1'b0	
2	Non-Zero-Length Status OUT Handshake (NZStsOUTHShk) The application can use this field to select the handshake the core sends on receiving a nonzero-length data packet during the OUT transaction of a control transfer's Status stage. 1'b1: Send a STALL handshake on a nonzero-length status OUT transaction and do not send the received OUT packet to the application. 1'b0: Send the received OUT packet to the application (zero-length or nonzero-length) and send a handshake based on the NAK and STALL bits for the endpoint in the Device Endpoint Control register.	1'b0	R_W
1:0	Device Speed (DevSpd) Indicates the speed at which the application requires the core to enumerate, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. 2'b00: High speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b01: Full speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 2'b10: Low speed (USB 1.1 transceiver clock is 6 MHz). If you select 6 MHz LS mode, you must do a soft reset. 2'b11: Full speed (USB 1.1 transceiver clock is 48 MHz)	2'b0	R_W

8.3.3.4.2 Device Control Register (DCTL)

Offset: 804h

Table 8-60: Device Control Register: DCTL

Field	Description	Reset	Access
31:12	Reserved	20'h0	
11	Power-On Programming Done (PWROnPrgDone) The application uses this bit to indicate that register programming is completed after a wake-up from Power Down mode.	1'b0	R_W
10	Clear Global OUT NAK (CGOUTNak) A write to this field clears the Global OUT NAK.	1'b0	WO
9	Set Global OUT NAK (SGOUTNak) A write to this field sets the Global OUT NAK. The application uses this bit to send a NAK handshake on all OUT endpoints. The application must set the this bit only after making sure that the Global OUT NAK Effective bit in the Core Interrupt Register (GINTSTS.GOUTNakEff) is cleared.	1'b0	WO
8	Clear Global Non-periodic IN NAK (CGNPInNak) A write to this field clears the Global Non-periodic IN NAK.	1'b0	WO
7	Set Global Non-periodic IN NAK (SGNPInNak) A write to this field sets the Global Non-periodic IN NAK.The application uses this bit to send a NAK handshake on all non-periodic IN endpoints. The core can also set this bit when a timeout condition is detected on a non-periodic endpoint in shared FIFO operation. The application must set this bit only after making sure that the Global IN NAK Effective bit in the Core Interrupt Register (GINTSTS.GINNakEff) is cleared.	1'b0	WO

© 2007 QuickLogic Corporation

Table 8-60: Device Control Register: DCTL (Continued)

Field	Description	Reset	Access
6:4	Test Control (TstCtl) 3'b000: Test mode disabled 3'b001: Test_J mode 3'b010: Test_K mode 3'b011: Test_SE0_NAK mode 3'b100: Test_Packet mode 3'b101: Test_Force_Enable Others: Reserved	3'b0	R_W
3	Global OUT NAK Status (GOUTNakSts) 1'b0: A handshake is sent based on the FIFO Status and the NAK and STALL bit settings. 1'b1: No data is written to the RxFIFO, irrespective of space availability. Sends a NAK handshake on all packets, except on SETUP transactions. All isochronous OUT packets are dropped.	1'b0	RO
2	Global Non-periodic IN NAK Status (GNPINNakSts) 1'b0: A handshake is sent out based on the data availability in the transmit FIFO. 1'b1: A NAK handshake is sent out on all non-periodic IN endpoints, irrespective of the data availability in the transmit FIFO.	1'b0	RO
1	Soft Disconnect (SftDiscon) The application uses this bit to signal the USB OTG core to do a soft disconnect. As long as this bit is set, the host does not see that the device is connected, and the device does not receive signals on the USB. The core stays in the disconnected state until the application clears this bit. The minimum duration for which the core must keep this bit set is specified in **Table 8-53**. 1'b0: Normal operation. When this bit is cleared after a soft disconnect, the core drives the phy_opmode_o signal on the UTMI+ to 2'b00, which generates a device connect event to the USB host. When the device is reconnected, the USB host restarts device enumeration. 1'b1: The core drives the phy_opmode_o signal on the UTMI+ to 2'b01, which generates a device disconnect event to the USB host.	1'b0	R_W
0	Remote Wakeup Signaling (RmtWkUpSig) When the application sets this bit, the core initiates remote signaling to wake up the USB host. The application must set this bit to instruct the core to exit the Suspend state. As specified in the USB 2.0 specification, the application must clear this bit 1–15 ms after setting it.	1'b0	R_W

Table 8-61 lists the minimum duration under various conditions for which the SoftDisconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect. To accommodate clock jitter, it is recommended that the application add some extra delay to the specified minimum duration.

Table 8-61: Minimum Duration for Soft Disconnect

Operating Speed	Device State	Minimum Duration
High speed	Suspended	1 ms + 2.5 µs
High speed	Idle	3 ms + 2.5 µs
High speed	Not Idle or Suspended (Performing transactions)	125 µs
Full speed/Low speed	Suspended	1 ms + 2.5 µs
Full speed/Low speed	Idle	2.5 µs
Full speed/Low speed	Not Idle or Suspended (Performing transactions)	2.5 µs

8.3.3.4.3 Device Status Register (DSTS)

Offset: 808h

This register indicates the status of the core with respect to USB-related events. It must be read on interrupts from Device All Interrupts (DAINT) register.

Table 8-62: Device Status Register: DSTS

Field	Description	Reset	Access
31:22	Reserved	10'h0	
21:8	Frame or Microframe Number of the Received SOF (SOFFN) When the core is operating at high speed, this field contains a microframe number. When the core is operating at full or low speed, this field contains a frame number.	14'h0	RO
7:4	Reserved	4'h0	
3	Erratic Error (ErrticErr) The core sets this bit to report any erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error) seen on the UTMI+ . Due to erratic errors, the USB OTG core goes into Suspended state and an interrupt is generated to the application with Early Suspend bit of the Core Interrupt register (GINTSTS.ErlySusp). If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover.	1'b0	RO
2:1	Enumerated Speed (EnumSpd) Indicates the speed at which the USB OTG core has come up after speed detection through a chirp sequence. 2'b00: High speed (PHY clock is running at 30 or 60 MHz) 2'b01: Full speed (PHY clock is running at 30 or 60 MHz) 2'b10: Low speed (PHY clock is running at 6 MHz) 2'b11: Full speed (PHY clock is running at 48 MHz) Low speed is not supported for devices using a UTMI+ PHY.	2'h0	RO
0	Suspend Status (SuspSts) In Device mode, this bit is set as long as a Suspend condition is detected on the USB. The core enters the Suspended state when there is no activity on the phy_line_state_i signal for an extended period of time. The core comes out of the suspend: When there is any activity on the phy_line_state_i signal When the application writes to the Remote Wakeup Signaling bit in the Device Control register (DCTL.RmtWkUpSig).	1'b0	RO

© 2007 QuickLogic Corporation

8.3.3.4.4 Device IN Endpoint Common Interrupt Mask Register (DIEPMSK)

Offset: 810h

This register works with each of the Device IN Endpoint Interrupt (DIEPINTn) registers for all endpoints to generate an interrupt per IN endpoint. The IN endpoint interrupt for a specific status in the DIEPINTn register can be masked by writing to the corresponding bit in this register. Status bits are masked by default.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Table 8-63: Device IN Endpoint Common Interrupt Mask Register: DIEPMSK

Field	Description	Reset	Access
31:9	Reserved	23'h0	
8	Fifo Underrun Mask (TxfifoUndrnMsk)	1'b0	R_W
7	Reserved.	1'b0	R_W
6	IN Endpoint NAK Effective Mask (INEPNakEffMsk)	1'b0	R_W
5	IN Token received with EP Mismatch Mask (INTknEPMisMsk)	1'b0	R_W
4	IN Token Received When TxFIFO Empty Mask (INTknTXFEmpMsk)	1'b0	R_W
3	Timeout Condition Mask (TimeOUTMsk) (Non-isochronous endpoints)	1'b0	R_W
2	Internal Bus Error Mask (AHBErrMsk)	1'b0	R_W
1	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	1'b0	R_W
0	Transfer Completed Interrupt Mask (XferComplMsk)	1'b0	R_W

8.3.3.4.5 Device OUT Endpoint Common Interrupt Mask Register (DOEPMSK)

Offset: 814h

This register works with each of the Device OUT Endpoint Interrupt (DOEPINTn) registers for all endpoints to generate an interrupt per OUT endpoint. The OUT endpoint interrupt for a specific status in the DOEPINTn register can be masked by writing into the corresponding bit in this register. Status bits are masked by default.

- Mask interrupt: 1'b0
- Unmask interrupt: 1'b1

Table 8-64: Device OUT Endpoint Common Interrupt Mask Register: DOEPMSK

Field	Description	Reset	Access
31:9	Reserved	23'h0	
8	OUT Packet Error Mask (OutPktErrMsk)	1'b0	R_W
7	Reserved	1'b0	
6	Back-to-Back SETUP Packets Received Mask (Back2BackSETup) Applies to control OUT endpoints only.	1'b0	R_W
5	Reserved	1'b0	
4	OUT Token Received when Endpoint Disabled Mask (OUTTknEPdisMsk) Applies to control OUT endpoints only.	1'b0	R_W
3	SETUP Phase Done Mask (SetUPMsk) Applies to control endpoints only.	1'b0	R_W

© 2007 QuickLogic Corporation

Table 8-64: Device OUT Endpoint Common Interrupt Mask Register: DOEPMSK (Continued)

Field	Description	Reset	Access
2	Internal Bus Error (AHBErrMsk)	1'b0	R_W
1	Endpoint Disabled Interrupt Mask (EPDisbldMsk)	1'b0	R_W
0	Transfer Completed Interrupt Mask (XferComplMsk)	1'b0	R_W

8.3.3.4.6 Device All Endpoints Interrupt Register (DAINT)

Offset: 818h

When a significant event occurs on an endpoint, a Device All Endpoints Interrupt register interrupts the application using the Device OUT Endpoints Interrupt bit or Device IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively). This is shown in Figure 8-3. There is one interrupt bit per endpoint, up to a maximum of 16 bits for OUT endpoints and 16 bits for IN endpoints. For a bidirectional endpoint, the corresponding IN and OUT interrupt bits are used. Bits in this register are set and cleared when the application sets and clears bits in the corresponding Device Endpoint-n Interrupt register (DIEPINTn/DOEPINTn).

Table 8-65: Device All Endpoints Interrupt Register: DAINT

Field	Description	Reset	Access
31:16	OUT Endpoint Interrupt Bits (OutEPInt) One bit per OUT endpoint: Bit 16 for OUT endpoint 0, bit 31 for OUT endpoint 15	16'h0	RO
15:0	IN Endpoint Interrupt Bits (InEpInt) One bit per IN Endpoint: Bit 0 for IN endpoint 0, bit 15 for endpoint 15	16'h0	RO

8.3.3.4.7 Device All Endpoints Interrupt Mask Register (DAINTMSK)

Offset: 81Ch

The Device Endpoint Interrupt Mask register works with the Device Endpoint Interrupt register to interrupt the application when an event occurs on a device endpoint. However, the Device All Endpoints Interrupt (DAINT) register bit corresponding to that interrupt is still set.

- Mask Interrupt: 1'b0
- Unmask Interrupt: 1'b1

Table 8-66: Device Endpoints Interrupt Mask Register: DAINTMSK

Field	Description	Reset	Access
31:16	OUT EP Interrupt Mask Bits (OutEpMsk) One per OUT Endpoint: Bit 16 for OUT EP 0, bit 31 for OUT EP 15	16'h0	R_W
15:0	IN EP Interrupt Mask Bits (InEpMsk) One bit per IN Endpoint: Bit 0 for IN EP 0, bit 15 for IN EP 15	16'h0	R_W

© 2007 QuickLogic Corporation

8.3.3.4.8 Device IN Token Sequence Learning Queue Read Register 1 (DTKNQR1)

Offset: 820h

This register is valid only in non-periodic Shared FIFO operation.

The depth of the IN Token Sequence Learning Queue is specified for Device Mode IN Token Sequence Learning Queue Depth during configuration. The queue is 4 bits wide to store the endpoint number. A read from this register returns the first 5 endpoint entries of the IN Token Sequence Learning Queue. When the queue is full, the new token is pushed into the queue and oldest token is discarded.

Table 8-67: Device IN Token Sequence Learning Queue Read Register 1: DTKNQR1

Field	Description	Reset	Access
31:8	Endpoint Token (EPTkn) Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 5 Bits [27:24]: Endpoint number of Token 4 ..... Bits [15:12]: Endpoint number of Token 1 Bits [11:8]: Endpoint number of Token 0	24'h0	RO
7	Wrap Bit (WrapBit) This bit is set when the write pointer wraps. It is cleared when the learning queue is cleared.	1'b0	RO
6:5	Reserved	2'h0	RO
4:0	IN Token Queue Write Pointer (INTknWPtr)	5'h0	RO

8.3.3.4.9 Device IN Token Sequence Learning Queue Read Register 2 (DTKNQR2)

Offset: 0824h

This register is valid only in shared non-periodic Shared FIFO operation.

A read from this register returns the next 8 endpoint entries of the learning queue.

Table 8-68: Device IN Token Sequence Learning Queue Register 2: DTKNQR2

Field	Description	Reset	Access
31:0	Endpoint Token (EPTkn) Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 13 Bits [27:24]: Endpoint number of Token 12 ..... Bits [7:4]: Endpoint number of Token 7 Bits [3:0]: Endpoint number of Token 6	32'h0	RO

8.3.3.4.10 Device IN Token Sequence Learning Queue Read Register 3 (DTKNQR3)

Offset: 0830h

This register is valid only in non-periodic Shared FIFO operation.

A read from this register returns the next 8 endpoint entries of the learning queue.

© 2007 QuickLogic Corporation

Table 8-69: Device IN Token Sequence Learning Queue Register 3: DTKNQR3

Field	Description	Reset	Access
31:0	Endpoint Token (EPTkn) Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 21 Bits [27:24]: Endpoint number of Token 20 ..... Bits [7:4]: Endpoint number of Token 15 Bits [3:0]: Endpoint number of Token 14	32'h0	RO

8.3.3.4.11 Device IN Token Sequence Learning Queue Read Register 4 (DTKNQR4)

Offset: 0834h

This register is valid only in non-periodic Shared FIFO operation.

A read from this register returns the last 8 endpoint entries of the learning queue.

Table 8-70: Device IN Token Sequence Learning Queue Register 4: DTKNQR2

Field	Description	Reset	Access
31:0	Endpoint Token (EPTkn) Four bits per token represent the endpoint number of the token: Bits [31:28]: Endpoint number of Token 29 Bits [27:24]: Endpoint number of Token 28 ..... Bits [7:4]: Endpoint number of Token 23 Bits [3:0]: Endpoint number of Token 22	32'h0	RO

8.3.3.4.12 Device VBUS Discharge Time Register (DVBUSDIS)

Offset: 0828h

This register specifies the VBUS discharge time after VBUS pulsing during SRP.

Table 8-71: Device VBUS Discharge Time Register: DVBUSDIS

Field	Description	Reset	Access
31:16	Reserved	16'h0	
15:0	Device VBUS Discharge Time (DVBUSDis) Specifies the VBUS discharge time after VBUS pulsing during SRP. This value equals: VBUS discharge time in PHY clocks / 1, 024 Depending on your VBUS load, this value can need adjustment.	60 MHz: 16'h17D7	R_W

8.3.3.4.13 Device VBUS Pulsing Time Register (DVBUSPULSE)

Offset: 082Ch

This register specifies the VBUS pulsing time during SRP.

© 2007 QuickLogic Corporation

Table 8-72: Device VBUS Pulsing Time Register (DVBUSPULSE)

Field	Description	Reset	Access
31:12	Reserved	20'h0	
11:0	Device VBUS Pulsing Time (DVBUSPulse) Specifies the VBUS pulsing time during SRP. This value equals: VBUS pulsing time in PHY clocks / 1, 024	60 MHz: 12'h5B8	R_W

8.3.3.4.14 Device IN Endpoint FIFO Empty Interrupt Mask Register: (DIEPEMPMSK)

Offset: 834 h

This register is not supported.

8.3.3.4.15 Device Control IN Endpoint 0 Control Register (DIEPCTL0)

Offset: 900h

This section describes the Control IN Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1–15.

Table 8-73: Device Control IN Endpoint 0 Control Register: DIEPCTL0

Field	Description	Reset	Access
31	Endpoint Enable (EPEna) Indicates that data is ready to be transmitted on the endpoint. The core clears this bit before setting any of the following interrupts on this endpoint: Endpoint Disabled Transfer Completed	1'b0	R_WS_SC
30	Endpoint Disable (EPDis) The application sets this bit to stop transmitting data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled Interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.	1'b0	R_WS_SC
29:28	Reserved	2'b0	
27	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for an endpoint after a SETUP packet is received on that endpoint.	1'b0	WO
26	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint.	1'b0	WO
25:22	TxFIFO Number (TxFNum) For Shared FIFO operation, this value is always set to 0, indicating that control IN endpoint 0 data is always written in the Non-Periodic Transmit FIFO. For Dedicated FIFO operation, this value is set to the FIFO number that is assigned to IN Endpoint 0.	4'h0	R_W

Table 8-73: Device Control IN Endpoint 0 Control Register: DIEPCTL0 (Continued)

Field	Description	Reset	Access
21	STALL Handshake (Stall) The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority.	1'b0	R_WS_SC
20	Reserved	1'b0	
19:18	Endpoint Type (EPType) Hardcoded to 00 for control.	2'h0	RO
17	NAK Status (NAKSts) Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status 1'b1: The core is transmitting NAK handshakes on this endpoint. When this bit is set, either by the application or core, the core stops transmitting data, even if there is data available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0	RO
16	Reserved	1'b0	
15	USB Active Endpoint (USBActEP) This bit is always set to 1, indicating that control endpoint 0 is always active in all configurations and interfaces.	1'b1	RO
14:11	Next Endpoint (NextEp) Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is not set. This field is not valid in Slave mode. **Note**: This field is valid only for Shared FIFO operations.	4'b0	R_W
10:2	Reserved	9'h0	
1:0	Maximum Packet Size (MPS) Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes 2'b11: 8 bytes	2'h0	R_W

Device Logical OUT Endpoint-Specific Registers

8.3.3.4.16 Device Control OUT Endpoint 0 Control Register (DOEPCTL0)

Offset: B00h

This section describes the Control OUT Endpoint 0 Control register. Nonzero control endpoints use registers for endpoints 1–15.

© 2007 QuickLogic Corporation

Table 8-74: Device OUT Endpoint 0 Control Register: DOEPCTL0

Field	Description	Reset	Access
31	Endpoint Enable (EPEna) Indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done Endpoint Disabled Transfer Completed **Note:** In DMA mode, this bit must be set for the core to transfer SETUP data packets into memory.	1'b0	R_WS_SC
30	Endpoint Disable (EPDis) The application cannot disable control OUT endpoint 0.	1'b0	RO
29:28	Reserved	2'b0	
27	Set NAK (SNAK) A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set bit on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.	1'b0	WO
26	Clear NAK (CNAK) A write to this bit clears the NAK bit for the endpoint.	1'b0	WO
25:22	Reserved	4'h0	
21	STALL Handshake (Stall) The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0	R_WS_SC
20	Snoop Mode (Snp) This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.	1'b0	R_W
19:18	Endpoint Type (EPType) Hardcoded to 2'b00 for control.	2'h0	RO
17	NAK Status (NAKSts) Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit, the core stops receiving data, even if there is space in the RxFIFO to accommodate the incoming packet. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0	RO
16	Reserved	1'b0	

Table 8-74: Device OUT Endpoint 0 Control Register: DOEPCTL0 (Continued)

Field	Description	Reset	Access
15	USB Active Endpoint (USBActEP) This bit is always set to 1, indicating that a control endpoint 0 is always active in all configurations and interfaces.	1'b1	RO
14:2	Reserved	13'h0	
1:0	Maximum Packet Size (MPS) The maximum packet size for control OUT endpoint 0 is the same as what is programmed in control IN Endpoint 0. 2'b00: 64 bytes 2'b01: 32 bytes 2'b10: 16 bytes 2'b11: 8 bytes	2'h0	RO

8.3.3.4.17 Device Endpoint-n Control Register (DIEPCTLn/DOEPCTLn)

Endpoint_number: $1 \leq n \leq 15$

Offset for IN endpoints: 900h + (Endpoint_number * 20h)

Offset for OUT endpoints: B00h + (Endpoint_number * 20h)

The application uses this register to control the behavior of each logical endpoint other than endpoint 0.

© 2007 QuickLogic Corporation

Table 8-75: Device Endpoint-n Control Register: DIEPCTLn/DOEPCTLn

Field	Description	Reset	Access
31	Endpoint Enable (EPEna) Applies to IN and OUT endpoints. For IN endpoints, this bit indicates that data is ready to be transmitted on the endpoint. For OUT endpoints, this bit indicates that the application has allocated the memory to start receiving data from the USB. The core clears this bit before setting any of the following interrupts on this endpoint: SETUP Phase Done (OUT only) Endpoint Disabled Transfer Completed **Note:** For control OUT endpoints in DMA mode, this bit must be set to be able to transfer SETUP data packets in memory.	1'b0	R_WS_SC
30	Endpoint Disable (EPDis) Applies to IN and OUT endpoints. The application sets this bit to stop transmitting/receiving data on an endpoint, even before the transfer for that endpoint is complete. The application must wait for the Endpoint Disabled interrupt before treating the endpoint as disabled. The core clears this bit before setting the Endpoint Disabled interrupt. The application must set this bit only if Endpoint Enable is already set for this endpoint.	1'b0	R_WS_SC
29	Set DATA1 PID (SetD1PID) Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA1. Set Odd (micro)frame (SetOddFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to odd (micro)frame.	1'b0	WO
28	Set DATA0 PID (SetD0PID) Applies to interrupt/bulk IN and OUT endpoints only. Writing to this field sets the Endpoint Data PID (DPID) field in this register to DATA0. Set Even (micro)frame (SetEvenFr) Applies to isochronous IN and OUT endpoints only. Writing to this field sets the Even/Odd (micro)frame (EO_FrNum) field to even (micro) frame.	1'b0	WO
27	Set NAK (SNAK) Applies to IN and OUT endpoints. A write to this bit sets the NAK bit for the endpoint. Using this bit, the application can control the transmission of NAK handshakes on an endpoint. The core can also set this bit for OUT endpoints on a Transfer Completed interrupt, or after a SETUP is received on the endpoint.	1'b0	WO
26	Clear NAK (CNAK) Applies to IN and OUT endpoints. A write to this bit clears the NAK bit for the endpoint.	1'b0	WO

© 2007 QuickLogic Corporation

Table 8-75: Device Endpoint-n Control Register: DIEPCTLn/DOEPCTLn (Continued)

Field	Description	Reset	Access
25:22	TxFIFO Number (TxFNum) *Shared FIFO Operation*—non-periodic endpoints must set this bit to zero. Periodic endpoints must map this to the corresponding Periodic xFIFO number. 4'h0: Non-Periodic TxFIFO Others: Specified Periodic TxFIFO.number **Note**: An interrupt IN endpoint can be configured as a nonperiodic endpoint for applications like mass storage.The core treats an IN endpoint as a non-periodic endpoint if the TxFNum field is set to 0. Otherwise, a separate periodic FIFO must be allocated for an interrupt IN endpoint, and the number of this FIFO must be programmed into the TxFNum field. Configuring an interrupt IN endpoint as a non-periodic endpoint saves the extra periodic FIFO area. *Dedicated FIFO Operation*—these bits specify the FIFO number associated with this endpoint. Each active IN endpoint should be programmed to a separate FIFO number.	4'h0	R_W
21	STALL Handshake (Stall) Applies to non-control, non-isochronous IN and OUT endpoints only. The application sets this bit to stall all tokens from the USB host to this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Only the application can clear this bit, never the core. Applies to control endpoints only. The application can only set this bit, and the core clears it, when a SETUP token is received for this endpoint. If a NAK bit, Global Non-periodic IN NAK, or Global OUT NAK is set along with this bit, the STALL bit takes priority. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0	R_W R_WS_SC
20	Snoop Mode (Snp) Applies to OUT endpoints only. This bit configures the endpoint to Snoop mode. In Snoop mode, the core does not check the correctness of OUT packets before transferring them to application memory.	1'b0	R_W
19:18	Endpoint Type (EPType) Applies to IN and OUT endpoints. This is the transfer type supported by this logical endpoint. 2'b00: Control 2'b01: Isochronous 2'b10: Bulk 2'b11: Interrupt	2'h0	R_W

© 2007 QuickLogic Corporation

Table 8-75: Device Endpoint-n Control Register: DIEPCTLn/DOEPCTLn (Continued)

Field	Description	Reset	Access
17	NAK Status (NAKSts) Applies to IN and OUT endpoints. Indicates the following: 1'b0: The core is transmitting non-NAK handshakes based on the FIFO status. 1'b1: The core is transmitting NAK handshakes on this endpoint. When either the application or the core sets this bit: The core stops receiving any data on an OUT endpoint, even if there is space in the RxFIFO to accommodate the incoming packet. For non-isochronous IN endpoints: The core stops transmitting any data on an IN endpoint, even if there data is available in the TxFIFO. For isochronous IN endpoints: The core sends out a zero-length data packet, even if there data is available in the TxFIFO. Irrespective of this bit's setting, the core always responds to SETUP data packets with an ACK handshake.	1'b0	RO
16	Endpoint Data PID (DPID) Applies to interrupt/bulk IN and OUT endpoints only. Contains the PID of the packet to be received or transmitted on this endpoint. The application must program the PID of the first packet to be received or transmitted on this endpoint, after the endpoint is activated. Applications use the SetD1PID and SetD0PID fields of this register to program either DATA0 or DATA1 PID. 1'b0: DATA0 1'b1: DATA1 Even/Odd (Micro)Frame (EO_FrNum) Applies to isochronous IN and OUT endpoints only. Indicates the (micro)frame number in which the core transmits/receives isochronous data for this endpoint. The application must program the even/odd (micro) frame number in which it intends to transmit/receive isochronous data for this endpoint using the SetEvnFr and SetOddFr fields in this register. 1'b0: Even (micro)frame 1'b1: Odd (micro)frame	1'b0	RO
15	USB Active Endpoint (USBActEP) Applies to IN and OUT endpoints. Indicates whether this endpoint is active in the current configuration and interface. The core clears this bit for all endpoints (other than EP 0) after detecting a USB reset. After receiving the SetConfiguration and SetInterface commands, the application must program endpoint registers accordingly and set this bit.	1'b0	R_W_SC
14:11	Next Endpoint (NextEp) Applies to non-periodic IN endpoints only. Indicates the endpoint number to be fetched after the data for the current endpoint is fetched. The core can access this field, even when the Endpoint Enable (EPEna) bit is low. This field is not valid in Slave mode operation. **Note**: This field is valid only for Shared FIFO operations.	4'b0	R_W
10:0	Maximum Packet Size (MPS) Applies to IN and OUT endpoints. The application must program this field with the maximum packet size for the current logical endpoint. This value is in bytes.	11'h0	R_W

© 2007 QuickLogic Corporation

8.3.3.4.18 Device Endpoint-n Interrupt Register (DIEPINTn/DOEPINTn)

Endpoint_number: $0 \leq n \leq 15$

Offset for IN endpoints: 908h + (Endpoint_number * 20h)

Offset for OUT endpoints: B08h + (Endpoint_number * 20h)

This register indicates the status of an endpoint with respect to USB- and Internal Bus-related events. It is shown in Figure 8-3. The application must read this register when the OUT Endpoints Interrupt bit or IN Endpoints Interrupt bit of the Core Interrupt register (GINTSTS.OEPInt or GINTSTS.IEPInt, respectively) is set. Before the application can read this register, it must first read the Device All Endpoints Interrupt (DAINT) register to get the exact endpoint number for the Device Endpoint-n Interrupt register. The application must clear the appropriate bit in this register to clear the corresponding bits in the DAINT and GINTSTS registers.

Table 8-76: Device Endpoint-n Interrupt Register: DIEPINTn/DOEPINTn

Field	Description	Reset	Access
31:9	Reserved	23'h0	
8	Fifo Underrun (TxfifoUndrn) Applies to IN endpoints Only. Not supported.	1'b0	R_W
7	Transmit FIFO Empty (TxFEmp) This bit is valid only for IN Endpoints This interrupt is asserted when the TxFIFO for this endpoint is either half or completely empty. The half or completely empty status is determined by the TxFIFO Empty Level bit in the Core Internal Bus Configuration register (GAHBCFG.NPTxFEmpLvl)).	1'b0	R_W
6	IN Endpoint NAK Effective (INEPNakEff) Applies to periodic IN endpoints only. Indicates that the IN endpoint NAK bit set by the application has taken effect in the core. This bit can be cleared when the application clears the IN endpoint NAK by writing to DIEPCTLn.CNAK. This interrupt indicates that the core has sampled the NAK bit set (either by the application or by the core). This interrupt does not necessarily mean that a NAK handshake is sent on the USB. A STALL bit takes priority over a NAK bit.	1'b0	RO
	Back-to-Back SETUP Packets Received (Back2BackSETup) Applies to Control OUT endpoints only. This bit indicates that the core has received more than three back-to-back SETUP packets for this particular endpoint.	1'b0	R_W
5	IN Token Received with EP Mismatch (INTknEPMis) Applies to non-periodic IN endpoints only. Indicates that the data in the top of the non-periodic TxFIFO belongs to an endpoint other than the one for which the IN token was received. This interrupt is asserted on the endpoint for which the IN token was received. For OUT endpoints, this bit is Reserved	1'b0	R_SS_WC

© 2007 QuickLogic Corporation

Table 8-76: Device Endpoint-n Interrupt Register: DIEPINTn/DOEPINTn (Continued)

Field	Description	Reset	Access
4	IN Token Received When TxFIFO is Empty (INTknTXFEmp) Applies to non-periodic IN endpoints only. Indicates that an IN token was received when the associated TxFIFO (periodic/non-periodic) was empty. This interrupt is asserted on the endpoint for which the IN token was received. OUT Token Received When Endpoint Disabled (OUTTknEPdis) Applies only to control OUT endpoints. Indicates that an OUT token was received when the endpoint was not yet enabled. This interrupt is asserted on the endpoint for which the OUT token was received.	1'b0	R_SS_WC
3	Timeout Condition (TimeOUT) Applies to non-isochronous IN endpoints in shared FIFO operation only. Indicates that the core has detected a timeout condition on the USB for the last IN token on this endpoint. SETUP Phase Done (SetUp) Applies to control OUT endpoints only. Indicates that the SETUP phase for the control endpoint is complete and no more back-to-back SETUP packets were received for the current control transfer. On this interrupt, the application can decode the received SETUP data packet.	1'b0	R_SS_WC
2	Internal Bus Error (AHBErr) Applies to IN and OUT endpoints. This is generated only in Internal DMA mode when there is an Internal Bus error during an Internal Bus read/write. The application can read the corresponding endpoint DMA address register to get the error address.	1'b0	R_SS_WC
1	Endpoint Disabled Interrupt (EPDisbld) Applies to IN and OUT endpoints. This bit indicates that the endpoint is disabled per the application's request.	1'b0	R_SS_WC
0	Transfer Completed Interrupt (XferCompl) Applies to IN and OUT endpoints. Indicates that the programmed transfer is complete on the Internal Bus as well as on the USB, for this endpoint.	1'b0	R_SS_WC

8.3.3.4.19 Device Endpoint 0 Transfer Size Register (DIEPTSIZ0/DOEPTSIZ0)

Offset for IN endpoints: 910h

Offset for OUT endpoints: B10h

The application must modify this register before enabling endpoint 0. Once endpoint 0 is enabled using Endpoint Enable bit of the Device Control Endpoint 0 Control registers (DIEPCTL0.EPEna/DOEPCTL0.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit.

Nonzero endpoints use the registers for endpoints 1–15.

Table 8-77: Device IN Endpoint 0 Transfer Size Register: DIEPTSIZ0

Field	Description	Reset	Access
31:20	Reserved	12'h0	
19	Packet Count (PktCnt) Indicates the total number of USB packets that constitute the Transfer Size amount of data for endpoint 0. This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO.	1'b0	R_W
18:7	Reserved	12'h0	
6:0	Transfer Size (XferSize) Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet from the external memory is written to the TxFIFO.	7'h0	R_W

© 2007 QuickLogic Corporation

Table 8-78: Device OUT Endpoint 0 Transfer Size Register: DOEPTSIZ0

Field	Description	Reset	Access
31	Reserved	1'b0	
30:29	SETUP Packet Count (SUPCnt) This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets	2'h0	R_W
28:20	Reserved	9'h0	
19	Packet Count (PktCnt) This field is decremented to zero after a packet is written into the RxFIFO.	1'b0	R_W
18:7	Reserved	12'h0	
6:0	Transfer Size (XferSize) Indicates the transfer size in bytes for endpoint 0. The core interrupts the application only after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.	7'h0	R_W

8.3.3.4.20 Device Endpoint-n Transfer Size Register (DIEPTSIZn/DOEPTSIZn)

Endpoint_number: $1 \leq n \leq 15$

Offset for IN endpoints: 910h + (Endpoint_number * 20h)

Offset for OUT endpoints: B10h + (Endpoint_number * 20h)

The application must modify this register before enabling the endpoint. Once the endpoint is enabled using Endpoint Enable bit of the Device Endpoint-n Control registers (DIEPCTLn.EPEna/DOEPCTLn.EPEna), the core modifies this register. The application can only read this register once the core has cleared the Endpoint Enable bit.

This register is used only for endpoints other than Endpoint 0.

© 2007 QuickLogic Corporation

Table 8-79: Device Endpoint-n Transfer Size Register: DIEPTSIZn/DOEPTSIZn

Field	Description	Reset	Access
31	Reserved	1'b0	
30:29	Multi Count (MC) Applies to IN endpoints only. For periodic IN endpoints, this field indicates the number of packets that must be transmitted per microframe on the USB. The core uses this field to calculate the data PID for isochronous IN endpoints. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets For non-periodic IN endpoints, this field is valid only in Internal DMA mode. It specifies the number of packets the core should fetch for an IN endpoint before it switches to the endpoint pointed to by the Next Endpoint field of the Device Endpoint-n Control register (DIEPCTLn.NextEp). Received Data PID (RxDPID) Applies to isochronous OUT endpoints only. This is the data PID received in the last packet for this endpoint. 2'b00: DATA0 2'b01: DATA1 2'b10: DATA2 2'b11: MDATA SETUP Packet Count (SUPCnt) Applies to control OUT Endpoints only. This field specifies the number of back-to-back SETUP data packets the endpoint can receive. 2'b01: 1 packet 2'b10: 2 packets 2'b11: 3 packets	2'b0	R_W RO R_W R_W
28:19	Packet Count (PktCnt) Indicates the total number of USB packets that constitute the Transfer Size amount of data for this endpoint. The power-on value is specified for Width of Packet Counters during configuration. IN Endpoints: This field is decremented every time a packet (maximum size or short packet) is read from the TxFIFO. OUT Endpoints: This field is decremented every time a packet (maximum size or short packet) is written to the RxFIFO.	10'h0	R_W
18:0	Transfer Size (XferSize) This field contains the transfer size in bytes for the current endpoint. The power-on value is specified for Width of Transfer Size Counters during configuration. The core only interrupts the application after it has exhausted the transfer size amount of data. The transfer size can be set to the maximum packet size of the endpoint, to be interrupted at the end of each packet. IN Endpoints: The core decrements this field every time a packet from the external memory is written to the TxFIFO. OUT Endpoints: The core decrements this field every time a packet is read from the RxFIFO and written to the external memory.	19'h0	R_W

8.3.3.4.21 Device Endpoint-n DMA Address Register (DIEPDMAn/DOEPDMAn)

Endpoint_number: $0 \leq n \leq 15$

© 2007 QuickLogic Corporation

Offset for IN endpoints: 914h + (Endpoint_number * 20h)

Offset for OUT endpoints: B14h + (Endpoint_number * 20h)

Table 8-80: Device Endpoint-n DMA Address Register: DIEPDMAn/DOEPDMAn

Field	Description	Reset	Access
31:0	DMA Address (DMAAddr) Holds the start address of the external memory for storing or fetching endpoint data. This register is incremented on every Internal Bus transaction. **Note:** For control endpoints, this address stores control OUT data packets as well as SETUP transaction data packets. If more than three SETUP packets are received back-to-back, the, the SETUP data packet in the memory is overwritten.	32'h0	R_W

8.3.3.4.22 Device IN Endpoint Transmit FIFO Status Register (DTXFSTSn)

Endpoint_number: $0 \leq n \leq 15$

Offset for IN endpoints: 918h + (Endpoint_number * 20h)

This read-only register contains the free space information for the Device IN endpoint TxFIFO.

Table 8-81: Device IN Endpoint Transmit FIFO Status Register: DTXFSTSn

Field	Description	Reset	Access
31:16	Reserved	1'b0	
15:0	IN Endpoint TxFIFO Space Avail (INEPTxFSpcAvail) Indicates the amount of free space available in the Endpoint TxFIFO. Values are in terms of 32-bit words. 16'h0: Endpoint TxFIFO is full 16'h1: 1 word available 16'h2: 2 words available 16'h*n*: *n* words available (where $0 \leq n \leq 32,768$) 16'h8000: 32,768 words available Others: Reserved	16'h0	RO

8.3.3.5 Power and Clock Gating Registers

8.3.3.5.1 Power and Clock Gating Control Register (PCGCCTL)

Offset: E00h

This register is available in Host and Device modes. The PwrClmp bit is available only if the OTG_EN_PWROPT parameter is set to 1 during core configuration. The application can use this register to control the core's power-down and clock gating features.

Because the CSR module is turned off during power-down, this registers is implemented in the Internal Bus Slave BIU module.

© 2007 QuickLogic Corporation

Table 8-82: Power and Clock Gating Control Register: PCGCCTL

Field	Description	Reset	Access
31:5	Reserved	27'h0	
4	PHY Suspended. (PhySuspended) Indicates that the PHY has been suspended. After the application sets the Stop Pclk bit (bit 0), this bit is updated once the PHY is suspended. Because the UTMI+ PHY suspend is controlled through a port, the UTMI+ PHY is suspended immediately after Stop Pclk is set. However, the ULPI PHY takes a few clocks to suspend, because the suspend information is conveyed through the ULPI protocol to the ULPI PHY.	1'b0	RO
3	Reset Power-Down Modules (RstPdwnModule) This bit is valid only in Partial Power-Down mode. The application sets this bit when the power is turned off. The application clears this bit after the power is turned on and the PHY clock is up.	1'b0	R_W
2	Power Clamp (PwrClmp) Not supported.	1'b0	R_W
1	Gate Hclk (GateHclk) The application sets this bit to gate hclk to modules other than the Internal Bus Slave and Master and wakeup logic when the USB is suspended or the session is not valid. The application clears this bit when the USB is resumed or a new session starts.	1'b0	R_W
0	Stop Pclk (StopPclk) The application sets this bit to stop the PHY clock (phy_clk) when the USB is suspended, the session is not valid, or the device is disconnected. The application clears this bit when the USB is resumed or a new session starts.	1'b0	R_W

© 2007 QuickLogic Corporation